



UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

Escola Politècnica Superior d'Enginyeria
de Manresa



Treball Final de Grau

Inclinòmetre Remot i Aplicació en una Motocicleta

Grau en Enginyeria de sistemes Tic

Curs 2018/2019

Autor: Daniel Trias Soler.

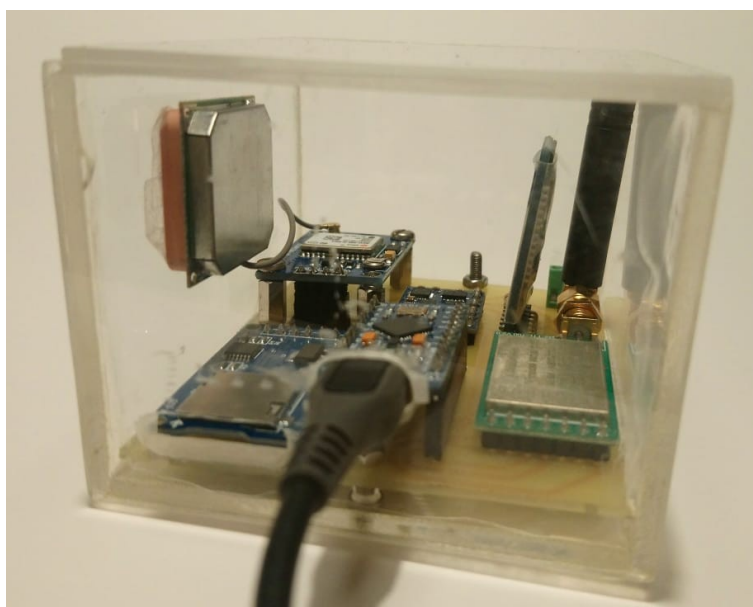
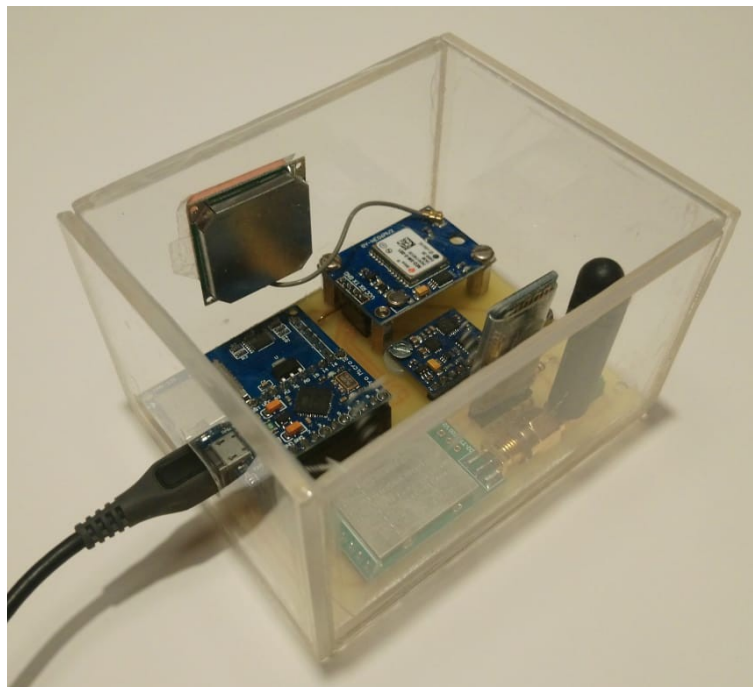
Director: Joan Gabriel Bergas Jane

Data: 11 / 10 / 2019

Localitat: Manresa

S'agraeix el suport i l'ajuda que s'ha facilitat des de la UPC Manresa en l'execució d'aquest projecte, així com la dels pares, amics i parella els quals han fet possible l'execució d'aquest projecte.

Agrair a l'empresa Servisò SCCL per facilitar el seu laboratori en la creació del prototip.



RESUM DEL PROJECTE (en català o castellà)

Aquest projecte consisteix en fabricar un inclinòmetre i transmetre les dades que aquest ens proporciona remotament.

Un cop fabricat l'aparell, s'estudiarà el seu comportament estàtic i el seu comportament dinàmic. En aquest cas, es vol estudiar el seu comportament en una motocicleta, per tal de poder veure les seves inclinacions a temps real. Particularment es vol aplicar aquest aparell per poder veure les inclinacions de la motocicleta quan aquesta estigui dins d'un circuit, per tant amb un abast limitat (aproximadament 1Km). A part també s'ha pensat amb una aplicació com a detector de caigudes per la motocicleta, d'aquesta forma també volem transmetre les dades via Bluetooth per que les pugui tractar un Smartphone.

En aquest projecte ens centrarem en el disseny de l'aparell, per tant en la seva fabricació, construcció, i tractament de dades obtingudes per aquest. Tanmateix, buscarem la forma de poder aplicar aquest aparell, per detectar les inclinacions d'una motocicleta.

Ens centrarem en buscar un microcontrolador adient pel processat de les dades de l'aparell, així com en trobar un sensor que ens permeti detectar les inclinacions dels objectes de varies formes. Farem proves estàtiques i dinàmiques amb aquests per poder estudiar els resultats i buscar la millor solució per poder-lo aplicar en un objecte en moviment. Seguidament estudiarem els sistemes LoRa i Bluetooth per tal de transmetre les dades remotament a aparells externs.

Project summary

This project is about building a gadget that measures the bending angle and that transmits the provided information remotely.

Once the device has been manufactured, its static and dynamic performance will be studied. In this case i want to study its performance on a motorbike in order to see the motorbike-inclination on real time. Particularly I want to apply it on a track to see how the motorbike bends. For this reason it has a limited range (one kilometer approximately). It is also thought of the possibility of using the gadget in another application, this is, to make this device to detect if any motorbike rider falls off. For this option, I would transmit the information by Bluetooth in order to enable a Smartphone to process it.

In this project, I will focus on the device design, its production, construction and the processing of the obtained data. I will search the way to apply this device to detect the inclinations of the motorbike.

I will concentrate in looking for a suitable microcontroller to process the information in the gadget and to find a sensor that allows to detect the inclinations of objects in different ways. I will do static and dynamic tests and study the results. My challenge is to find the best option to apply it on a dynamic object. After all, I will study the LoRa and Bluetooth systems to find the best way to transmit the pieces of information remotely to external devices.

ÍNDEX

| | |
|--|---------|
| 1. INTRODUCCIÓ | pag 5. |
| 1.1 Objectiu | pag 5. |
| 1.2 Finalitat del Projecte. | pag 5. |
| 1.3 Abast. | pag 6. |
| 1.4 Antecedents. | pag 6 . |
| 2. COS DEL PROJECTE. | pag 11. |
| 2.1 Hardware. | pag 11. |
| 2.1.1 El microcontrolador. | pag 11. |
| 2.1.2 El Sensor GY-85. | pag 15. |
| 2.1.3 Mòdul de memòria microSD. | pag 20. |
| 2.1.4 Mòdul de transmissió de dades Bluetooth. | pag 22. |
| 2.1.5 Mòdul de transmissió de dades LoRa AS32-TTL-100. | pag 24. |
| 2.1.6 Mòdul GPS GY-NEO6MV2. | pag 26. |
| 2.1.7 Disseny del circuit imprès. | pag 27. |
| 2.2 Software. | pag 38. |
| 2.2.1 Microcontrolador Pro micro. | pag 38. |
| 2.2.2 Lectura de les dades del sensor GY-85. | pag 40. |
| 2.2.3 Guardat de les dades en una targeta microSD. | pag 45. |
| 2.2.4 Detecció dels angles. | pag 49. |
| 2.2.5 Transmissió de les dades amb el mòdul LoRa. | pag 56. |
| 2.2.6 Transmissió de les dades amb el mòdul Bluetooth. | pag 60. |
| 2.2.7 El mòdul GPS. | pag 67. |

| | |
|---|----------|
| 2.3 Proves i treball de camp. | pag 76 |
| 2.3.1 Prova 1 Detecció de la inclinació absoluta i relativa. | pag 77. |
| 2.3.2 Prova 2 Detecció de la inclinació en estàtic, detecció 10°. | pag 81. |
| 2.3.3 Prova 3 Detecció de la inclinació en estàtic, detecció de 90° i retorn a 0°. | pag 89. |
| 2.3.4 Prova 4 Detecció de la inclinació en estàtic, filtre 98% giroscopi 2% acceleròmetre. | pag 90. |
| 2.3.5 Prova 5 Determinació dels valors del filtre complementari, 85% giroscopi, 15% acceleròmetre. | pag 92. |
| 2.3.6 Prova 6 Utilització del sistema en la motocicleta, determinació dels angles a baixa velocitat en l'aparcament de casa. | pag 94. |
| 2.3.7 Prova 7 caiguda en parat. | pag 98. |
| 2.3.8 Prova 8 dinàmica amb velocitat, traçat d'una rotonda, canvi de sentit. | pag 100. |
| 2.3.9 Prova 9 Localització de la caiguda. | pag 101. |
| 2.4 Fitxa tècnica de l'aparell. | pag 104. |
| 3. ESTUDI ECONÒMIC. | pag 105. |
| 4. CONCLUSIONS. | pag 107. |
| 4.1 Conclusions sobre l'aparell fabricat. | pag 107. |
| 4.2 Futures línies de treball. | pag 108. |
| 5 BIBLIOGRAFIA. | pag 109. |
| 6 ANNEXOS | pag 110. |

6.1 Annex 1. Muntatge del prototip a la motocicleta. _____ pag 110.

ÍNDEX DE FIGURES

| | |
|--|---------|
| Figura 1 → Angles de Tait-Bryan. | pag 6. |
| Figura 2 → Exemple de l'angle d'inclinació en una motocicleta. | pag 7. |
| Figura 3 → Espirals i cercles. | pag 8. |
| Figura 4 → Microcontrolador AtMega 32U4. | pag 11. |
| Figura 5 → Pin-out. | pag 13. |
| Figura 6 → Eixos de l'acceleròmetre. | pag 16. |
| Figura 7 → Especificacions de l'acceleròmetre ADXL345. | pag 17. |
| Figura 8 → Orientació de la Sensibilitat dels Eixos i Polaritat de Rotació. | pag 18. |
| Figura 9 → Especificacions del Giroscopi ITG3205. | pag 18. |
| Figura 10 → Cablejat GY-85 al microcontrolador BUS I2C. | pag 19. |
| Figura 11 → Mòdul microSD. | pag 20. |
| Figura 12 → Cablejat mòdul microSD al microcontrolador BUS SPI. | pag 21. |
| Figura 13 → Cablejat mòdul Bluetooth Hc-05 al microcontrolador Bus Serial. | pag 23. |
| Figura 14 → Cablejat del mòdul LoRa AS32-TTL-100 al microcontrolador bus Serial. | pag 24. |
| Figura 15 → Cablejat del mòdul GPS GY-NEO6MV2 al microcontrolador bus Serial. | pag 27. |
| Figura 16 → Programa KiCad. | pag 28. |
| Figura 17 → Esquema elèctric de la placa de circuit imprès. | pag 29. |
| Figura 18 → Placa PCB amb els components. | pag 32. |
| Figura 19 → Placa PCB amb dues cares. | pag 32. |
| Figura 20 → Placa PCB cara de la Impressió. | pag 33. |
| Figura 21 → Fotolit per impressió. | pag 34. |

| | |
|--|---------|
| Figura 22 → Dissolució de NaOH Sosa Càustica. | pag 35. |
| Figura 23 → Immersió de la placa en el rebel·lat. | pag 35. |
| Figura 24 → Efecte del rebel·lat. | pag 35. |
| Figura 25 → Rebel·lat finalitzat. | pag 35. |
| Figura 26 → Preparat Ariston per atacar el coure. | pag 36. |
| Figura 27 → Placa de circuit dins la solució àcida. | pag 36. |
| Figura 28 → Placa finalitzada. | pag 37. |
| Figura 29 → Arduino, Archivo -> Preferencias. | pag 38. |
| Figura 30 → Gestor de targetes. | pag 39. |
| Figura 31 → Gestor de targetes, busca de targetes Sparkfun. | pag 39. |
| Figura 32 → Selecció de la placa. | pag 40. |
| Figura 33 → Tipus de placa. | pag 40. |
| Figura 34 → Acceleracions sense escalat 4G. | pag 44. |
| Figura 35 → Especificacions del giroscopi per fer l'escalat. | pag 45. |
| Figura 36 → Sistema 2D. | pag 49. |
| Figura 37 → Esquema per la configuració del mòdul Bluetooth. | pag 61. |
| Figura 38 → Instrument de mesura d'angles. | pag 77. |
| Figura 39 → Mesura absoluta dels 0° amb l'acceleròmetre. | pag 79. |
| Figura 40 → Valors del prototip en graus. | pag 79. |
| Figura 41 → Angle relatiu a partir de l'acceleròmetre. | pag 80. |
| Figura 42 → Mesura de 10° amb l'acceleròmetre. | pag 81. |
| Figura 43 → Mesura de 10° amb l'acceleròmetre. | pag 81. |
| Figura 44 → Mesura de -10° amb l'acceleròmetre. | pag 82. |
| Figura 45 → Mesura de -10° amb l'acceleròmetre. | pag 82. |

| | |
|---|---------|
| Figura 46 → Mesura de 0° amb el giroscopi. | pag 83. |
| Figura 47 → Mesura de 10° amb el giroscopi intent 1. | pag 84. |
| Figura 48 → Mesura de 10° amb el giroscopi intent 2. | pag 84. |
| Figura 49 → Mesura de 10° amb el giroscopi intent 3. | pag 84 |
| Figura 50 → Retorn de 10° a 0° amb el giroscopi. | pag 84 |
| Figura 51 → Mesura de 10° acceleròmetre i giroscopi. | pag 85. |
| Figura 52 → Gràfica 1, inclinació 10° acceleròmetre. | pag 87. |
| Figura 53 → Gràfica 2, inclinació 10° giroscopi. | pag 88. |
| Figura 54 → Gràfica 1 acceleròmetre i filtre +90° -> 0° -> -90°-> 0°. | pag 89. |
| Figura 55 → Gràfica 2 giroscopi i filtre +90° -> 0° -> -90°-> 0°. | pag 90. |
| Figura 56 → Gràfica 1 acceleròmetre i filtre +80° -> 0°. | pag 91. |
| Figura 57 → Gràfica 2 giroscopi i filtre +80° -> 0°. | pag 91. |
| Figura 58 → Gràfica 1 acceleròmetre i filtre +80° -> 0°. | pag 92. |
| Figura 59 → Gràfica 2 giroscopi i filtre +80° -> 0°. | pag 93. |
| Figura 60 → Soroll de l'acceleròmetre 4G amb l'aparell sobre la motocicleta sense moviment però amb el motor en marxa. | pag 95. |
| Figura 61 → Soroll de l'acceleròmetre 16G amb l'aparell sobre la motocicleta sense moviment però amb el motor en marxa. | pag 95. |
| Figura 62 → Acceleració i frenada amb la motocicleta. Acceleròmetre a 4G. | pag 96. |
| Figura 63 → Acceleració i frenada amb la motocicleta. Acceleròmetre a 16G. | pag 96. |
| Figura 64 → Doble corba a la dreta, acceleròmetre a 4G i giroscopi. | pag 97. |
| Figura 65 → Corba a la dreta, acceleròmetre a 16G i giroscopi. | pag 98. |
| Figura 66 → Valors de l'acceleròmetre en la caiguda, velocitat=0Km/h. | pag 98. |
| Figura 67 → Valors del giroscopi en la caiguda, velocitat= 0Km/h. | pag 99. |

Figura 68 → Gràfica d'acceleracions i inclinacions en el traçat de una rotonda, canvi de sentit. _____ pag 100.

Figura 69 → Gràfica de la velocitat de rotació i inclinacions en el traçat de una rotonda, canvi de sentit. _____ pag 100.

Figura 70 → Localització GPS en ciutat. _____ pag 102.

Figura 71 → Localització GPS en el Google Maps. _____ pag 102.

1. INTRODUCCIÓ

1.1. Objectiu.

L'objectiu d'aquest projecte, és la creació d'un aparell que ens permeti detectar d'una forma remota la inclinació d'objectes que estiguin en moviment, com pot ser una motocicleta. Serà amb una d'aquestes màquines en la qual hi buscarem una aplicació directe.

Aquest objectiu final l'aconseguirem a partir d'aconseguir petits objectius.

Objectius:

1. Elecció del Hardware i creació de la placa de circuit imprès que incorpora tots els dispositius. D'aquesta forma aconseguim una base on desenvolupar el nostre algoritme.
2. Desenvolupar un algoritme a partir del prototip Hardware que ens permeti detectar les inclinacions en estàtic.
3. Desenvolupar un algoritme que ens permeti enviar les dades de forma remota, a partir dels dispositius de radiocomunicació escollits.
4. Incorporar el dispositiu en una motocicleta.
5. Estudiar el comportament del dispositiu i l'algoritme en la motocicleta.
6. Fer les modificacions que calgui en l'algoritme per tal de detectar les inclinacions de la motocicleta amb el màxim de precisió possible.

1.2. Finalitat del Projecte.

La finalitat principal d'aquest projecte, és la creació d'un aparell que ens permeti detectar els graus d'inclinació d'un objecte qualsevol. Concretament, volem detectar la inclinació d'objectes els quals estan en moviment, ja que és una tasca que no poden desenvolupar un gran nombre d'aparells.

La aplicació prevista de l'aparell, és mesurar els graus d'inclinació d'una motocicleta, per tal de detectar si aquesta ha patit una caiguda.

A la vegada es pretén poder aconseguir les inclinacions de la motocicleta en temps real en un radi aproximat d'un quilòmetre. D'aquesta forma l'aparell també tindria una aplicació com a telemetria en un circuit de velocitat.

1.3. Abast.

El que volem aconseguir en aquest projecte, és la construcció d'un aparell capaç de detectar les inclinacions en una motocicleta, i enviar aquestes inclinacions mitjançant la radiofreqüència, per tal d'aconseguir una lectura de les dades a temps real.

A part també volem aconseguir detectar quan la motocicleta ha patit una caiguda, i poder enviar un missatge fins a un Smartphone per tal de tractar les dades posteriorment.

1.4 Antecedents.

La tasca que ens hem plantejat és la de fabricar un inclinòmetre. Un inclinòmetre és un aparell capaç de detectar les inclinacions d'un objecte. En concret un objecte pot tindre varis tipus de inclinacions, aquestes inclinacions es poden determinar de varies formes, ja sigui mitjançant els angles d'Euler, les matrius de rotació o els quaternions. Tot i que la més estesa i intuïtiva és la dels angles de navegació, o angles de Tait-Bryan, en els quals la orientació es representa com tres rotacions ortogonals en tornavís, l'eix X (Roll), l'eix Y (Pitch), i l'eix Z(Yaw).

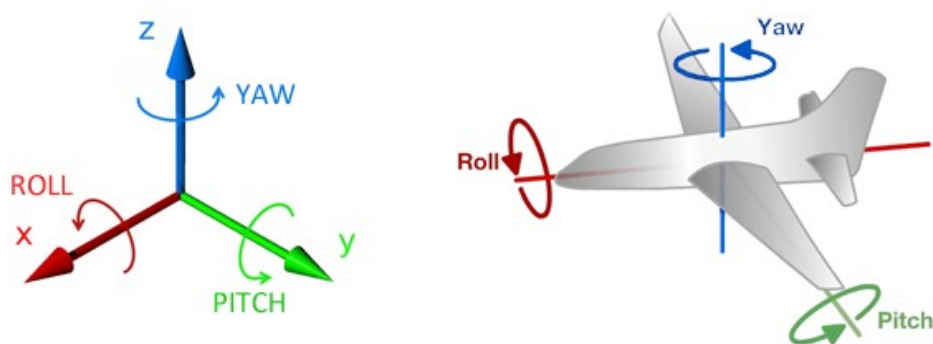


Figura 1: Angles de Tait-Bryan

Per detectar les inclinacions del nostre aparell utilitzarem un IMU (Inertial Measurement Units). En concret el model que utilitzarem serà un GY-85. Aquest sensor incorpora un Acceleròmetre de 3 eixos ADXL345, un giroscopi de 3 eixos ITG3205, i un magnetòmetre de 3 eixos HMC5883L, tot i que aquest últim no el tindrem en compte en el projecte. En concret ens centrarem en detectar les inclinacions a partir de la unió dels valors proporcionats per l'acceleròmetre i el giroscopi, aplicant un filtre capaç d'aconseguir el millor de cada un d'ells.

Tot i això en el nostre cas no necessitarem detectar tants eixos, amb 1 en tenim prou, en concret ens interessa l'eix de les X (Roll), ja que aquest ens indicarà la inclinació de l'aparell en la motocicleta.

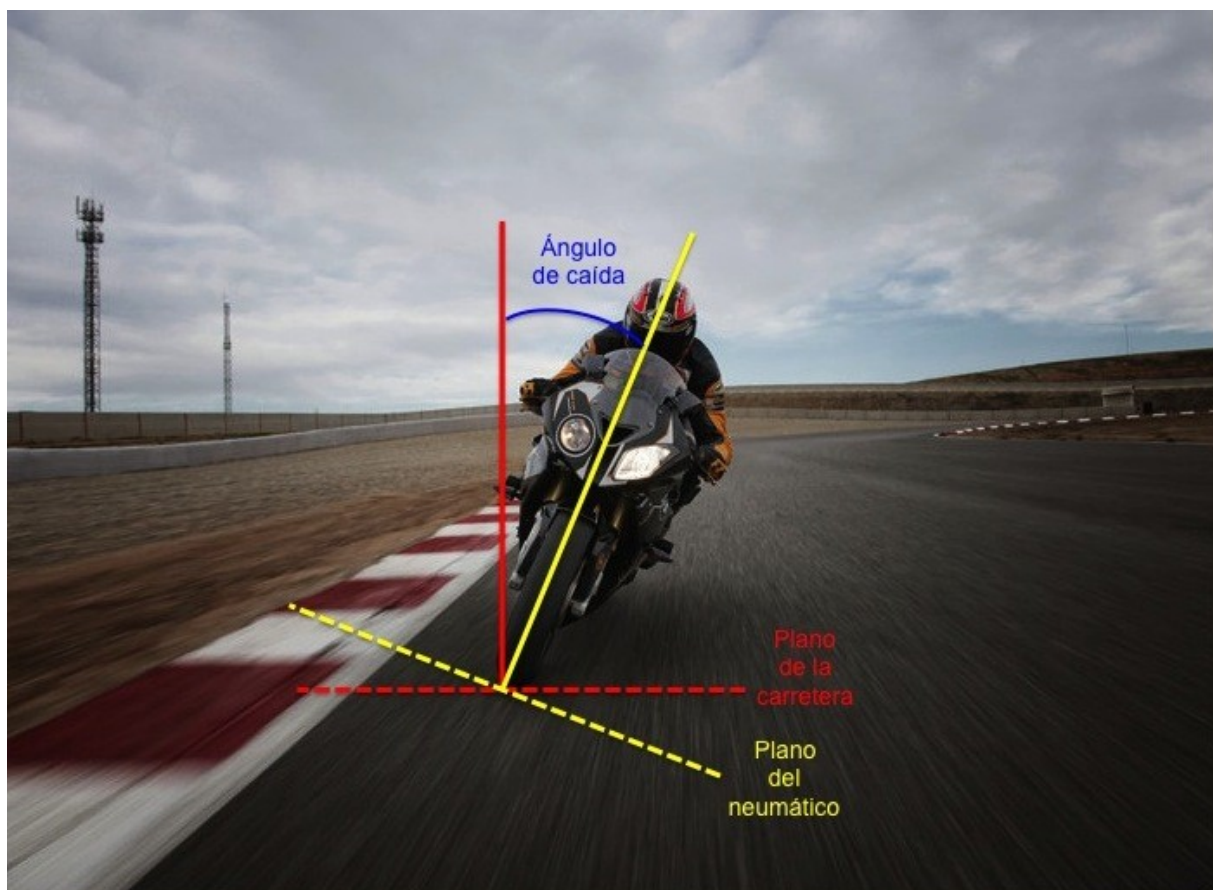


Figura 2: Exemple de l'angle d'inclinació en una motocicleta.

Per fer un desenvolupament del projecte eficaç, desenvoluparem el projecte per fases. Aquestes fases són principalment, el desenvolupament del Hardware, i el desenvolupament del Software, el qual s'ha d'efectuar d'una forma determinada pel correcte desenvolupament del projecte.

El hardware el desenvoluparem en forma de cercles, els quals es centren en el desenvolupament per anells. Cada un d'aquests anells significarà una evolució en el projecte. Primerament ens recolzarem de la placa de prototipat pel desenvolupament del nostre prototip, ja que en aquesta, és molt fàcil fer petites modificacions sense que el cost del projecte es dispari i sense que aquestes modificacions suposin un cost de temps molt alt. Un cop tinguem tot el sistema Hardware clar, passarem al disseny de una placa de circuit imprès per tal de donar-li una forma més definitiva al nostre projecte, aquest pas ens aportarà, rigidesa estructural i una major seguretat a l'hora d'efectuar les proves sobre elements mòbils.

A la vegada que estem dissenyant el Hardware, també dissenyarem el Software de l'aparell per etapes. Tot i que en aquest cas ens decantarem per un desenvolupament en forma d'espiral. Mentre anem desenvolupant el Hardware, a la vegada anirem desenvolupant el Software d'aquest de forma in-crescendo. Efectuant i testejant el funcionament de petits programes per cada un dels nostres elements Hardware que incorporarem en l'aparell.

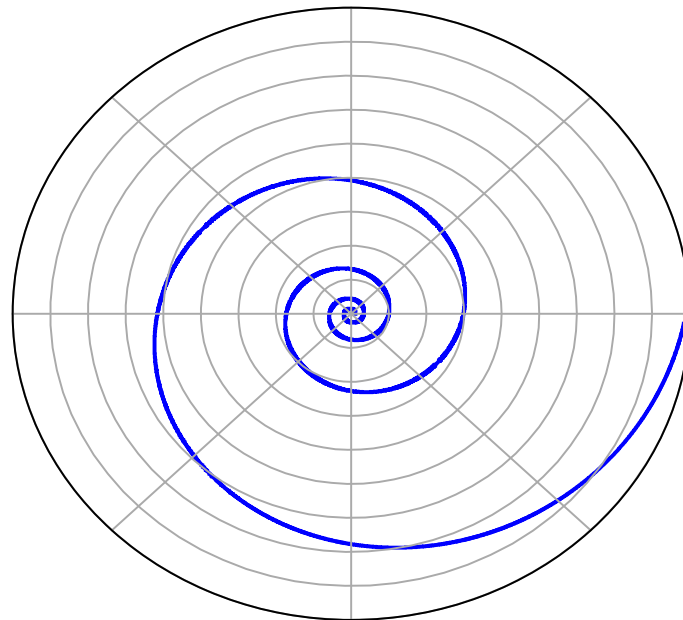


Figura 3: Espirals i cercles.

Podem apreciar una imatge dels models mencionats en la figura 3. Espirals i cercles. En aquesta figura, podem apreciar com es desenvoluparà el nostre sistema, sent l'espiral de color blau el desenvolupament del Software, i els cercles de color gris el desenvolupament del Hardware.

Els coneixements que ens permetran efectuar aquest projecte, són els adquirits en el grau d'Enginyeria TIC. Dels quals aplicarem els coneixements de programació en C, programació en Octave, disseny de plaques de circuit imprès, comunicacions de sistemes de radiofreqüència, i coneixements de electricitat bàsics.

2. COS DEL PROJECTE.

2.1. Hardware.

L'aparell que es proposa dissenyar en aquest projecte, és una junció d'elements encarregats de fer cada un d'ells una feina molt específica. En aquest apartat s'explica la missió de cada un d'ells i per que són seleccionats per el nostre projecte.

2.1.1 El Microcontrolador.

El microcontrolador que s'ha escollit pel projecte, és un Atmega 32U4, el qual es troba encapsulat dins d'una placa coneguda comercialment amb el nom de Pro micro. Podem veure una foto d'aquest en la figura 4.

El microcontrolador és el denominat cervell del projecte, és l'encarregat de comunicar-se i gestionar tots els perifèrics i mòduls externs que l'envolten, a l'hora, s'encarrega de fer càlculs amb totes les dades que aquests li proporcionen i d'actuar en conseqüència depenent del programa que se li carregui.



Figura 4: Microcontrolador AtMega 32U4

Aquesta, no és una versió oficial d'Arduino. Pot venir en 2 formats, el format de 3.3V i el format de 5V. Serà aquest últim en el que ens centrarem en el nostre cas.

2.1.1.1 Especificacions:

| | | |
|----------------------|-------------------------------|--|
| Arquitectura | 8 bits | Mida dels nombres amb els quals el nostre microcontrolador pot operar en cada flanc de rellotge. |
| Memòria flash | 32 KBytes (4KB Bootloader) | Memòria de Programa, espai que té el microcontrolador per guardar el programa que li carreguem, si en traiem l'alimentació, aquesta, es queda amb les dades emmagatzemades anteriorment. |
| SRAM | 2 KBytes | Memòria volàtil del microcontrolador, és el mateix que una memòria Ram en un ordinador, en aquest espai el microcontrolador carrega les dades un cop inicialitzat el programa. Si aquest perd l'alimentació la informació d'aquesta es perdrà. |
| EEPROM | 1KBytes | Memòria que serveix per emmagatzemar dades després del Reset. |
| Freqüència màxima | 16MHz a 4.5V | Velocitat a la qual opera el microcontrolador per segon. |
| Voltatge de Operació | 2.7V a 5.5V | Voltatge amb el qual podem alimentar el nostre microcontrolador. |
| Suport USB natiu | Micro USB | Ens permet no necessitar cap tipus de convertidor per poder-lo connectar a l'ordinador. |
| Intensitat per pin | 10mA | Es pot extreure per cada pin del microcontrolador 10mA. Es recomana no excedir els 5mA. |
| Mides de la placa | 17mm x 33mm | La placa en la que està integrada el microcontrolador té una mida de 17mm d'amplada i 33mm de llargada. Tenint en compte les seves mides, és ideal per integrar en projectes més grans. |
| Creadors de la placa | Sparkfun | https://learn.sparkfun.com/tutorials/pro-micro--fio-v3-hookup-guide/all |

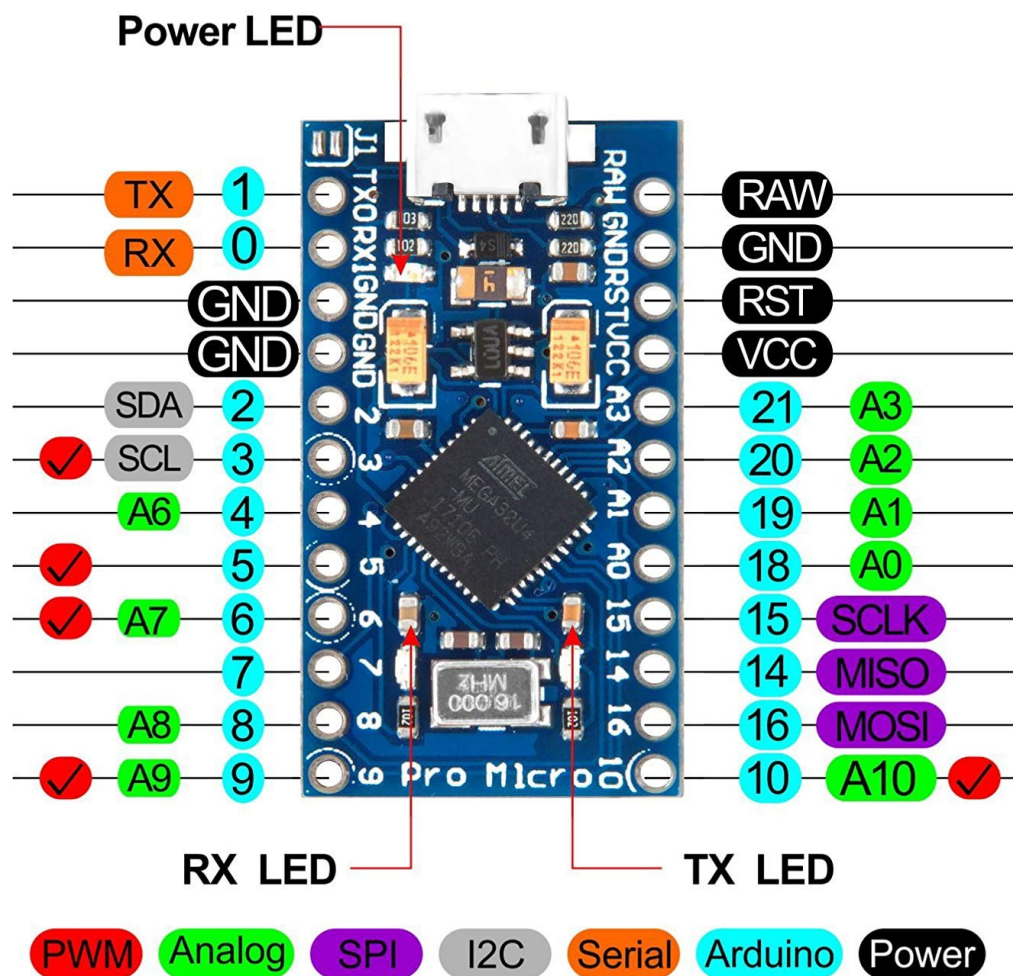


Figura 5: Pin-out.

En la figura 5, podem apreciar el detall del pin-out del microcontrolador. Si ens fixem en la llegenda, es pot observar d'esquerra a dreta, que disposem de sortides PWM, entrades Analògiques, bus SPI i I2C, port Serial, el nom que se li dona a cada sortida mitjançant la IDE Arduino i els pins d'alimentació.

Anem a explicar les característiques principals d'aquests pins per tal de conèixer més a fons el microcontrolador.

2.1.1.2 Pins del microcontrolador:

- **Alimentació.**

| | |
|-----|--|
| RAW | Pin d'alimentació externa en la versió de 5V aquesta pot ser des de 5V fins a 12V. |
| GND | És el 0V de la nostre placa. Elèctricament és defineix com a pin de potencial 0 o referència. |
| RST | Pin per reiniciar el microcontrolador externament, en aquest cas és d'activació negada, per tant, per tal d'efectuar un reset en el nostre microcontrolador aquest s'haurà de connectar a 0V o GND. |
| VCC | Pin que pot proporcionar alimentació a altres perifèrics que vulguem connectar al voltant del nostre microcontrolador, en el nostre cas, aquest té una alimentació de 5V i una intensitat de sortida màxima de 500mA. Aquest pin també es pot utilitzar per alimentar la placa del microcontrolador a partir de una font Dc de 5V. |

- **Pins especials.**

| | |
|-------------------|---|
| Entrada Analògica | Els pins 4, 6, 8, 9, 10, 18, 19, 10 i 21 són pins que permeten entrada analògica. Concretament aquests disposen d'un convertidor AC/DC de 10 bits, 1024 valors disponibles. |
| Sortida PWM | Els pins 3, 5, 6, 9 i 10 són pins amb possibilitat de modulació digital PWM de 8 bits, fins a 255 valors disponibles. |
| SPI | Els pins 15 SCLK, 14 MISO, i 16 MOSI, formen Bus de comunicació SPI (Serial Peripheral Interface), aquest port s'utilitza per la intercomunicació amb perifèrics que vulguem posar al voltant del nostre microcontrolador, com per exemple un lector de targetes microSD. Per la intercomunicació amb aquests dispositius necessitarem un pin digital extra per poder habilitar l'esclau. |
| I2C | Els pins 2 SDA, i 3 SCL, formen el Bus de comunicació I2C (Inter-integrated Circuit). Aquest bus de comunicació s'utilitza per la intercomunicació de dades entre un mestre i varis esclaus, els quals estan tots posats en una adreça diferent. El mestre és qui comença |

| | |
|----------------|---|
| | sempre la comunicació, en tal cas, l'esclau contesta amb les dades corresponents demandades per el mestre. |
| SPP (UART TTL) | Aquest bus s'utilitza per la comunicació Serial TTL amb altres mòduls a partir dels seus pins TX i RX, s'ha de tenir en compte que per aquest tipus de comunicació el canal ha d'anar creuat RX del microcontrolador amb TX del mòdul i TX del microcontrolador amb RX del mòdul. Hi ha varis mòduls que permeten aquest tipus de comunicació com pot ser el modul Bluetooth Hc-06. |

S'ha escollit aquest microcontrolador pel projecte degut a les seves dimensions reduïdes, la seva bona connectivitat, ja que disposa d'un segon port Serial, a part dels ports I2C i SPI, per la seva facilitat a la hora de programar-lo, ja que el podem programar amb la mateixa IDE d'Arduino fent alguna petita modificació, així com la seva facilitat per poder-lo integrar en projectes més grans.

2.1.2 El Sensor GY-85.

El sensor que porta integrat el projecte per tal de poder detectar les inclinacions, és el GY-85. Aquest sensor és un IMU de 9 DOF (9 eixos: 3 per l'acceleròmetre, 3 pel giroscopi i 3 pel magnetòmetre).

Aquesta placa incorpora els següents components:

| | | |
|-----------------------|-----------|---|
| Regulador de voltatge | 5V a 3.3V | Permet poder alimentar la placa del sensor amb 5V directament. Aquests els podem extreure directament del microcontrolador. |
| Acceleròmetre | ADXL345 | És un sensor de 3 eixos el qual ens permet detectar les acceleracions que pateix la placa en els eixos X,Y,Z. |
| Giroscopi | ITG3205 | És un sensor de 3 eixos el qual ens permet detectar la velocitat angular que pateix la placa en els eixos X,Y, Z. |
| Magnetòmetre | HMC5883L | És un sensor de 3 eixos el qual ens permet detectar la orientació de la placa, ja que és capaç de detectar els camps magnètics, com ara el camp magnètic terrestre. |

Pel projecte només utilitzarem la part de l'acceleròmetre i del giroscopi, per tant seguidament s'estudiaran aquests 2 amb més detall.

2.1.2.1 Acceleròmetre ADXL345:

ACCELERATION SENSITIVITY

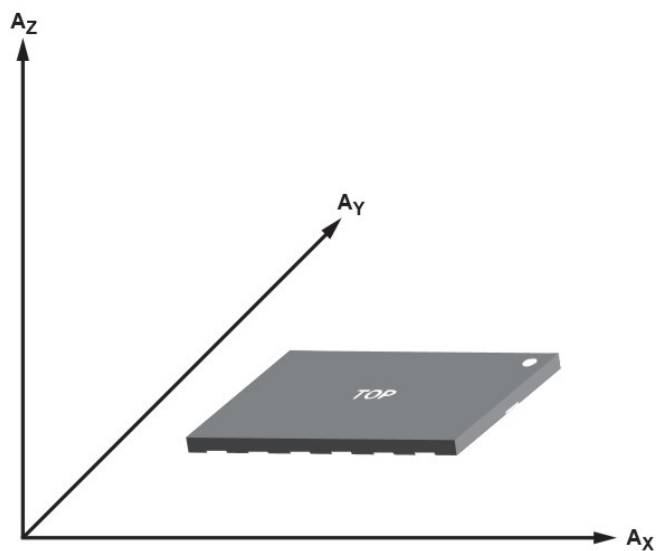


Figura 6: Eixos de l'acceleròmetre

Un acceleròmetre en termes generals és un dispositiu electromecànic que pot mesurar les acceleracions estàtiques (gravetat) i dinàmiques (moviments i vibracions), en cada un dels eixos de treball, tenint com a principi de funcionament les lleis de Newton.

Aquest dispositiu se sol utilitzar per mesurar les acceleracions lineals, les inclinacions i el posicionament de cossos.

L'ADXL345 és un sistema de baix consum de potència, que mesura acceleracions en els tres eixos(x, y, z), es pot programar per realitzar mesures estàtiques o dinàmiques. La configuració estàtica és utilitzada per calcular la inclinació d'objectes on ha sigut connectat el sensor. En les mesures de temps dinàmic, s'eliminen les acceleracions generades per moviments o xocs.

La resolució de sortida en el cas de mesures estàtiques és de 4mG/LSB, el qual vol dir que el canvi del bit menys representatiu mostra una variació de 4 mili gravetats, permetent avaluar angles de inclinacions inferiors a 1°.

En la següent taula es poden veure les especificacions del sensor:

| Parameter | Test Conditions | Min | Typ | Max | Unit |
|--|--|-------|-----------------|----------------|---------|
| SENSOR INPUT | Each axis | | | | |
| Measurement Range | User selectable | | ±2, ±4, ±8, ±16 | | g |
| Nonlinearity | Percentage of full scale | | ±0.5 | | % |
| Inter-Axis Alignment Error | | | ±0.1 | | Degrees |
| Cross-Axis Sensitivity ² | | | ±1 | | % |
| OUTPUT RESOLUTION | Each axis | | | | |
| All g Ranges | 10-bit resolution | | 10 | | Bits |
| ±2 g Range | Full resolution | | 10 | | Bits |
| ±4 g Range | Full resolution | | 11 | | Bits |
| ±8 g Range | Full resolution | | 12 | | Bits |
| ±16 g Range | Full resolution | | 13 | | Bits |
| SENSITIVITY | Each axis | | | | |
| Sensitivity at X _{OUT} , Y _{OUT} , Z _{OUT} | ±2 g, 10-bit or full resolution | 232 | 256 | 286 | LSB/g |
| Scale Factor at X _{OUT} , Y _{OUT} , Z _{OUT} | ±2 g, 10-bit or full resolution | 3.5 | 3.9 | 4.3 | mg/LSB |
| Sensitivity at X _{OUT} , Y _{OUT} , Z _{OUT} | ±4 g, 10-bit resolution | 116 | 128 | 143 | LSB/g |
| Scale Factor at X _{OUT} , Y _{OUT} , Z _{OUT} | ±4 g, 10-bit resolution | 7.0 | 7.8 | 8.6 | mg/LSB |
| Sensitivity at X _{OUT} , Y _{OUT} , Z _{OUT} | ±8 g, 10-bit resolution | 58 | 64 | 71 | LSB/g |
| Scale Factor at X _{OUT} , Y _{OUT} , Z _{OUT} | ±8 g, 10-bit resolution | 14.0 | 15.6 | 17.2 | mg/LSB |
| Sensitivity at X _{OUT} , Y _{OUT} , Z _{OUT} | ±16 g, 10-bit resolution | 29 | 32 | 36 | LSB/g |
| Scale Factor at X _{OUT} , Y _{OUT} , Z _{OUT} | ±16 g, 10-bit resolution | 28.1 | 31.2 | 34.3 | mg/LSB |
| Sensitivity Change Due to Temperature | | | ±0.01 | | %/°C |
| 0 g BIAS LEVEL | Each axis | | | | |
| 0 g Output for X _{OUT} , Y _{OUT} | | -150 | ±40 | +150 | mg |
| 0 g Output for Z _{OUT} | | -250 | ±80 | +250 | mg |
| 0 g Offset vs. Temperature for x-, y-Axes | | | ±0.8 | | mg/°C |
| 0 g Offset vs. Temperature for z-Axis | | | ±4.5 | | mg/°C |
| NOISE PERFORMANCE | | | | | |
| Noise (x-, y-Axes) | Data rate = 100 Hz for ±2 g, 10-bit or full resolution | | <1.0 | | LSB rms |
| Noise (z-Axis) | Data rate = 100 Hz for ±2 g, 10-bit or full resolution | | <1.5 | | LSB rms |
| OUTPUT DATA RATE AND BANDWIDTH | User selectable | | | | |
| Measurement Rate ³ | | 6.25 | | 3200 | Hz |
| SELF-TEST⁴ | Data rate ≥ 100 Hz, 2.0 V ≤ V _S ≤ 3.6 V | | | | |
| Output Change in x-Axis | | 0.20 | | 2.10 | g |
| Output Change in y-Axis | | -2.10 | | -0.20 | g |
| Output Change in z-Axis | | 0.30 | | 3.40 | g |
| POWER SUPPLY | | | | | |
| Operating Voltage Range (V _S) | | 2.0 | 2.5 | 3.6 | V |
| Interface Voltage Range (V _{DDIO}) | V _S ≤ 2.5 V | 1.7 | 1.8 | V _S | V |
| | V _S ≥ 2.5 V | 2.0 | 2.5 | V _S | V |
| Supply Current | Data rate > 100 Hz | | 145 | | μA |
| | Data rate < 10 Hz | | 40 | | μA |
| Standby Mode Leakage Current | | | 0.1 | 2 | μA |
| Turn-On Time ⁵ | Data rate = 3200 Hz | | 1.4 | | ms |
| TEMPERATURE | | | | | |
| Operating Temperature Range | | -40 | | +85 | °C |
| WEIGHT | | | | | |
| Device Weight | | | 20 | | mg |

Figura 7: Especificacions de l'acceleròmetre ADXL345. (extret de la referència ADXL345 datasheet).

2.1.2.2 Giroscopi ITG3205:

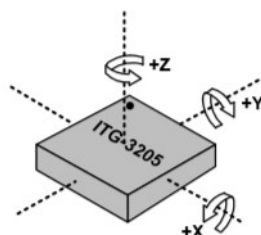


Figura 8: Orientació de la Sensibilitat dels Eixos i Polaritat de Rotació.

Un giroscopi és un element electromecànic utilitzat per mesurar velocitat angular en $^{\circ}/s$ o RPS (revolucions per segon). La velocitat de rotació la mesura sobre els eixos de treball; és utilitzat per sistemes d'equilibri, en navegació, mesura d'angles, entre altres. Treballa amb tecnologia MEMS (Sistema micro electromecànics).

El giroscopi ITG3205 té 3 conversos analògics digitals que converteixen les dades de gir de cada un dels eixos en valors digitals de 16 bits. Mesura freqüències de fins a 400Khz en el mode de transmissió ràpida. La sensibilitat és de 2000 $^{\circ}/s$, a escala completa. L'escala és programable per l'usuari des dels registres.

Podem veure les característiques del sensor a la taula següent.

| Parameter | Conditions | Min | Typical | Max | Unit | Note |
|---|--|------|-------------|-----|-------------------|------|
| GYRO SENSITIVITY | | | | | | |
| Full-Scale Range | FS_SEL=3 | | ± 2000 | | $^{\circ}/s$ | 4 |
| Gyro ADC Word Length | | | 16 | | Bits | 3 |
| Sensitivity Scale Factor | FS_SEL=3 | | 14.375 | | LSB/ $^{\circ}/s$ | 3 |
| Sensitivity Scale Factor Tolerance | 25°C | -6 | | +6 | % | 1 |
| Sensitivity Scale Factor Variation Over Temperature | 0°C to 55°C | | ± 10 | | % | 2 |
| Nonlinearity | Best fit straight line; 25°C | | 0.2 | | % | 6 |
| Cross-Axis Sensitivity | | | 2 | | % | 6 |
| GYRO ZERO-RATE OUTPUT (ZRO) | | | | | | |
| Initial ZRO Tolerance | | | ± 60 | | $^{\circ}/s$ | 1 |
| ZRO Variation Over Temperature | 0°C to 55°C | | ± 40 | | $^{\circ}/s$ | 2 |
| Power-Supply Sensitivity (1-10Hz) | Sine wave, 100mVpp; VDD=2.2V | | 0.2 | | $^{\circ}/s$ | 5 |
| Power-Supply Sensitivity (10 - 250Hz) | Sine wave, 100mVpp; VDD=2.2V | | 0.2 | | $^{\circ}/s$ | 5 |
| Power-Supply Sensitivity (250Hz - 100kHz) | Sine wave, 100mVpp; VDD=2.2V | | 4 | | $^{\circ}/s$ | 5 |
| Linear Acceleration Sensitivity | Static | | 0.1 | | $^{\circ}/s/g$ | 6 |
| GYRO NOISE PERFORMANCE | | | | | | |
| Total RMS noise | FS_SEL=3 100Hz LPF (DLPFCFG=2) | | 0.7 | | $^{\circ}/s$ -rms | 1 |
| GYRO MECHANICAL FREQUENCIES | | | | | | |
| X-Axis | | 30 | 33 | 36 | kHz | 1 |
| Y-Axis | | 27 | 30 | 33 | kHz | 1 |
| Z-Axis | | 24 | 27 | 30 | kHz | 1 |
| Frequency Separation | Between any two axes | 1.7 | | | kHz | 1 |
| GYRO START-UP TIME | | | | | | |
| ZRO Settling | DLPFCFG=0 to $\pm 1^{\circ}/s$ of Final | | 50 | | ms | 6 |
| TEMPERATURE SENSOR | | | | | | |
| Range | | | -30 to +585 | | $^{\circ}C$ | 2 |
| Sensitivity | | | 280 | | LSB/ $^{\circ}C$ | 2 |
| Temperature Offset | 35°C | | -13,200 | | LSB | 1 |
| Initial Accuracy | 35°C | | TBD | | $^{\circ}C$ | |
| Linearity | Best fit straight line (-30°C to +585°C) | | ± 1 | | $^{\circ}C$ | 2, 5 |
| TEMPERATURE RANGE | | | | | | |
| Specified Temperature Range | | 0-40 | | 585 | $^{\circ}C$ | |

Figura 9: Especificacions del Giroscopi ITG3205

2.1.2.3 Comunicació I2C amb la placa GY-85 :

Per poder comunicar el microcontrolador i el sensor, utilitzarem el protocol I2C (Inter-integrated circuit), aquest és un protocol de comunicacions desenvolupat per l'empresa Philips. Aquest protocol permet intercomunicar múltiples dispositius fins un element final el qual és el microcontrolador.

En aquest protocol les dades passen per un bus de dades compost per dues línies una denominada com a SCL i una segona denominada SDA. En aquest protocol existeix comunicació entre un mestre en aquest cas el microcontrolador i varis esclaus que en aquest cas són l'acceleròmetre ADXL345 i el giroscopi ITG3205. En aquest tipus de comunicació cada dispositiu té una direcció diferent i el mestre és el que controla la comunicació entre els diferents dispositius, concretament el mestre inicia la comunicació amb l'esclau corresponent, indicant que el bus està ocupat per la transmissió de les dades amb aquell esclau en concret.

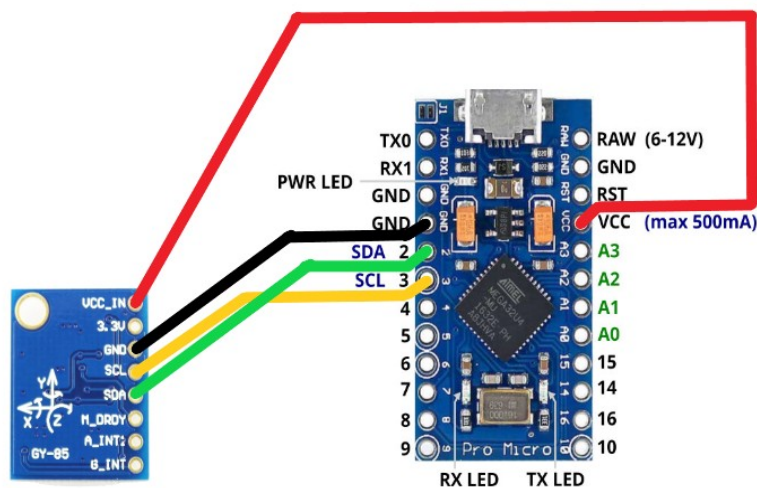


Figura 10: Cablejat GY-85 al microcontrolador BUS I2C.

En la figura 9 podem veure les línies de connexió entre el sensor Gy-85 i el microcontrolador. Aprofitem l'alimentació que ens proporciona la pròpia placa del microcontrolador per alimentar la placa del sensor. A la vegada que connectem el bus I2C a partir de les potes 2 SDA i 3 SCL del microcontrolador. Com es pot apreciar a la figura 9, aquests pins de connexió ja es troben marcats en la placa del sensor GY-85. D'aquesta

forma per poder connectar el bus I2C només hem de connectar la pota SDA del microcontrolador amb el pin SDA del sensor, i la pota SCL del microcontrolador amb el pin SCL del sensor. Un cop efectuades aquestes connexions ja es pot provar el sensor a partir de la comunicació I2C.

2.1.3 Mòdul de memòria microSD.

Per que el dispositiu pugui guardar dades, s'incorpora un mòdul per poder llegir i escriure targetes micro SD. Aquest mòdul permetrà poder escriure les dades que es rebin del sensor en una targeta de memòria micro SD. Un cop guardades les dades, es poden tractar aquestes posteriorment a partir de programes com Octave. Aquest permet generar gràfiques, les quals permeten veure millor el que està succeint en cada moment en el sistema.

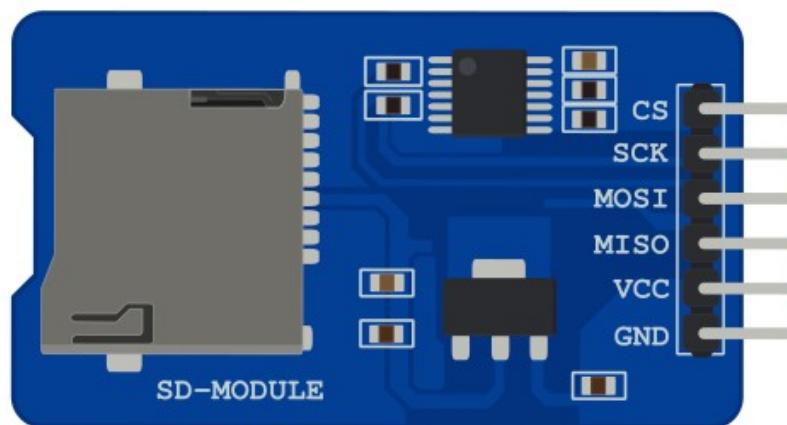


Figura 11: Mòdul microSD

La comunicació amb aquesta placa es porta a terme a partir del bus SPI ja que és l'únic que està disponible en aquesta placa.

La tensió d'alimentació de la placa l'extraurem del microcontrolador com en els altres mòduls. Alimentarem la placa a partir del pin VCC amb 5 V.

Es poden utilitzar targetes dels tipus SD o SDSC (Standard Capacity) o SDHC (High Capacity), però no SDXC (Extended Capacity). Aquestes hauran d'estar formatades en sistema d'arxius FAT16 o FAT32.

Utilitzar una targeta micro SD en un microprocessador té la avantatge de proporcionar una memòria molt gran per que aquest pugui guardar dades. Segons com i quin tipus de dades haguem d'emmagatzemar aquesta es podria considerar quasi infinita. A més, aquesta és un

tipus de memòria no volàtil, el que vol dir que quan tallem l'alimentació les dades emmagatzemades fins aleshores es conserven en la targeta.

El principal desavantatge d'aquest sistema, és que suposa una important carrega de treball per un microcontrolador com el que s'està utilitzant. Ja que suposarà una ocupació molt gran de la memòria Flash i de la memòria dinàmica.

2.1.3.1 Comunicació SPI amb el mòdul targeta microSD:

El Bus SPI (Serial Peripheral Interface) és un estàndard de comunicacions, utilitzat principalment per la transferència de informació entre circuits integrats en equips electrònics que permeti un flux de bits serie regulats per un rellotge (comunicació síncrona).

Aquest Bus està format per una línia de rellotge, una de dades d'entrada, una de dades de sortida, i un pin de chip select, el qual controla la connexió o desconexió del dispositiu amb el que un desitja comunicar-se.

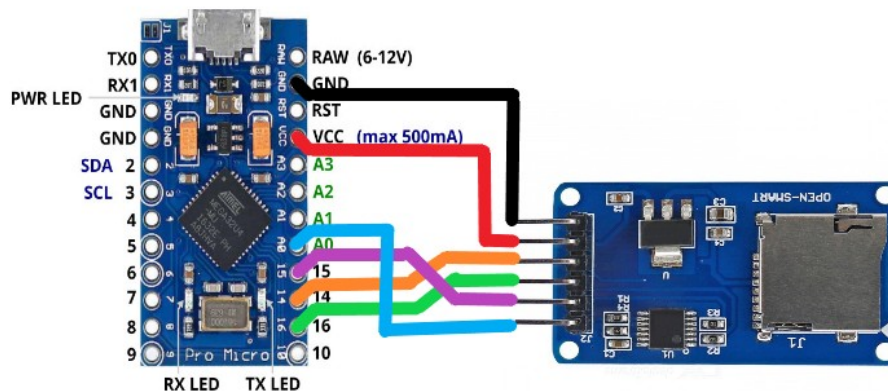


Figura 12: Cablejat mòdul microSD al microcontrolador BUS SPI

En la figura 11, es pot apreciar el muntatge del cablejat que s'efectua entre el microcontrolador i el mòdul microSD a partir del Bus SPI, en aquest cas format per els pins del microcontrolador 15, 14, 16. Com es pot apreciar en la figura 11, també s'aprofita l'alimentació que ens proporciona el microcontrolador a partir del pin VCC. El pin A0 el qual no pertany al bus SPI, és seleccionat com a sortida chip select per poder activar i desactivar el mòdul quan es desitgi.

Els pins de comunicació són els següents:

- SCLK (Clock) : És el pols que marca la sincronització. En la figura 11 es pot apreciar com el pin 15 i la línia violeta.
- MOSI (Màster Output Slave Input): És la sortida de dades del màster i la entrada de dades de l'esclau. En la figura 11 es pot apreciar com el pin 16 i la línia verda.
- MISO (Màster Input Slave Output): És la sortida de dades de l'esclau i entrada de dades del màster. En la figura 11 es pot apreciar com el pin 14 i la línia taronja.
- SS/Select: És utilitzat per seleccionar un esclau i poder-se comunicar amb ell. En la figura 11 es pot apreciar com el pin A0 i la línia blava.

L'avantatge principal d'aquest bus és la comunicació full dúplex que ofereix una gran velocitat de transmissió de dades.

El principal desavantatge, és que consumeix un major nombre de pins que no pas el bus I2C, i que l'activació de cada esclau es fa a partir de línies específiques per tant com més esclaus tinguem més línies haurem de ocupar com a chip select.

2.1.4 Mòdul de transmissió de dades Bluetooth.

Bluetooth és una especificació industrial per Xarxes sense fils d'Àrea Personal (WPAN), el qual fa possible la transmissió de dades entre diferents dispositius mitjançant un enllaç per radiofreqüència en la banda ISM dels 2.4 GHz. Els principals objectius que es pretenen aconseguir amb aquesta norma són:

- Facilitar les comunicacions entre equips mòbils.
- Eliminar els cables i connectors entre els dispositius.
- Oferir la possibilitat de crear petites xarxes sense fils i facilitar la sincronització de dades entre equips personals.

Aquest protocol està dissenyat especialment per dispositius de baix consum, que requereixen un curt abast d'emissió.

En aquest projecte utilitzarem el mòdul Bluetooth HC-05, el qual és igual al seu germà HC-06 amb la diferència que el mòdul HC-06 només pot actuar com a esclau i el mòdul HC-05 es pot utilitzar com a màster o esclau depenent de la seva configuració. Tot i això nosaltres utilitzarem el mòdul en mode esclau.

La capacitat de canal d'aquest dispositiu és la Versió 2.0+EDR el qual té un ample de banda de 3 Mbit/s. I segueix l'estàndard de comunicacions IEEE 802.15.

Aquest mòdul Bluetooth té una potència de transmissió de 4dBm el qual el qualifica com a classe 2 amb una potència màxima de 2.5 mW, i un abast aproximat de 10 metres.

Ens comunicarem amb el mòdul Bluetooth a partir de la comunicació SPP (Serial Port Protocol) TX, RX el qual ens proporciona el microcontrolador com a Serial1.

La configuració d'aquest mòdul la veurem en l'apartat Software 2.2.6, i s'efectua a partir de comandes AT.

En quan a l'alimentació del mòdul s'utilitza l'alimentació que ens ofereix el microcontrolador a partir del seu pin VCC de 5V.

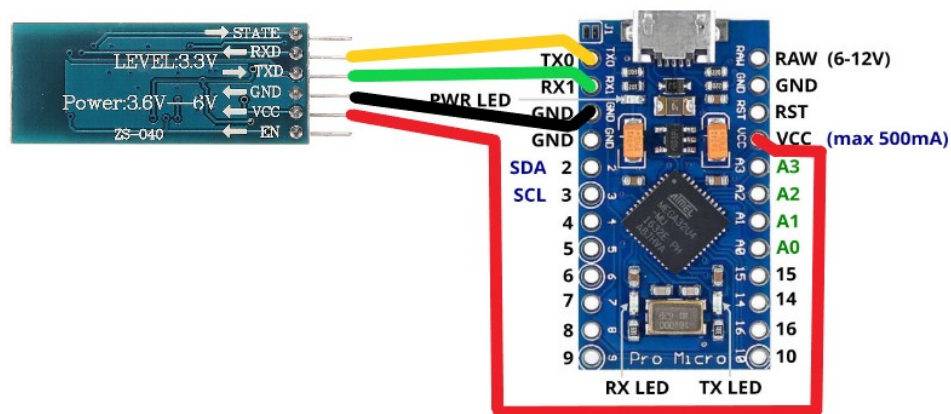


Figura 13: Cablejat mòdul Bluetooth Hc-05 al microcontrolador Bus Serial.

Es pot apreciar en la figura 12, el cablejat del mòdul Bluetooth i el microcontrolador a partir del Bus Serial. En aquesta figura veiem que el bus Serial es connecta de forma creuada; El pin TX del microcontrolador va connectat al pin RX del mòdul Bluetooth i el pin RX del microcontrolador va connectat al pin TX del mòdul Bluetooth. Aquesta connexió és així, ja que en el Bus Serial el pin TX és el de sortida de dades i el pin RX és el d'entrada de dades amb dos costats. Per tant la sortida de dades d'un sistema ha d'anar connectat a l'entrada de dades de l'altre sistema.

2.1.5 Mòdul de transmissió de dades LoRa AS32-TTL-100.

El mòdul de transmissió de dades LoRa (Long Range) AS32-TTL-100 forma part de les denominades xarxes LPWAN (Low-Power Wide Area Network) o xarxes de baix consum i àrea estesa. Aquesta tecnologia té grans avantatges per projectes de Internet de les Coses o IoT (Internet of Things).

LoRa és una tecnologia de comunicació sense fils que utilitza una modulació denominada Chirp Spread Spectrum, o CSS patentada per Semtech, el qual s'utilitza amb comunicacions militars i especials des de fa dècades. El gran avantatge de la mateixa és que pot aconseguir comunicacions de llarga distància (normalment de Kilòmetres), i té una gran solidesa davant de les interferències.

Avantatges:

- Alta tolerància a les interferències.
- Alta sensibilitat per rebre dades (-168dB).
- Baix Consum 100mW, amb 5 V 110mA.
- Llarg abast (3000m).
- Baixa transferència: de 2,4Kbits/s en un espai obert.
- Connexió punt a punt.
- Freqüència de treball de 433Mhz.
- Interfície de comunicació SPP (Serial Port Protocol).
- Búffer TX/RX de 256 Bytes en paquets de 58 Bytes.

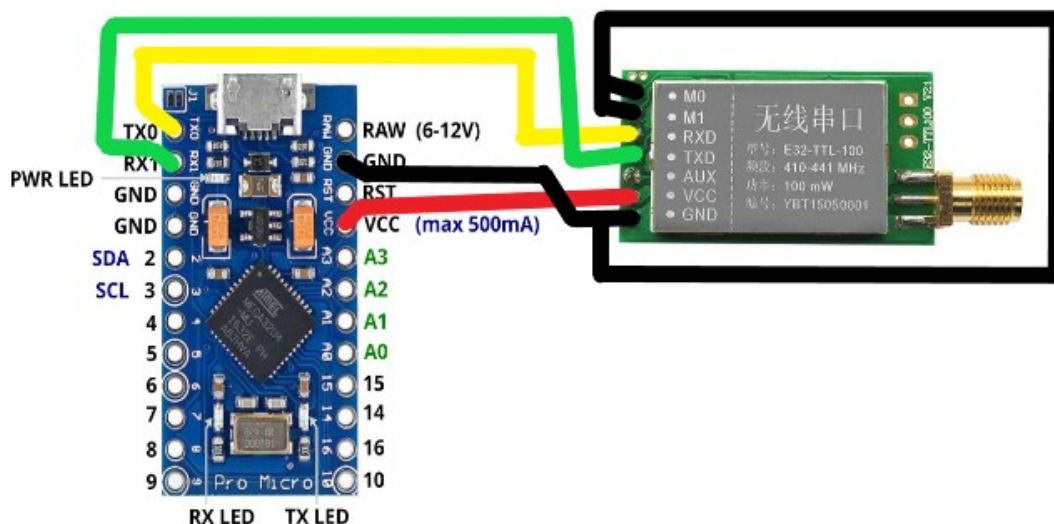


Figura 14: Cablejat del mòdul LoRa AS32-TTL-100 al microcontrolador bus Serial

En la figura 13 es pot veure les connexions entre el mòdul LoRa AS32-TTL-100 i el microcontrolador. Es pot apreciar que l'alimentació s'extreu del propi microcontrolador i que la transferència de dades es fa mitjançant el port Serial1, tenint en compte que el pin Tx del microcontrolador va connectat al pin Rx del mòdul de transmissió LoRa i el pin Rx del microcontrolador va connectat al pin Tx del mòdul de transmissió LoRa.

2.1.5.1 Definició dels pins del mòdul AS32-TTL-100:

| Pin No. | Pin | Aplicació |
|---------|-----|--|
| 1 | M0 | Entrada: es combina amb la entrada M1 per establir el mode de treball. |
| 2 | M1 | Entrada: es combina amb la entrada M0 per establir el mode de treball. |
| 3 | RX | Entrades TTL connectades amb una sortida TX externa. |
| 4 | TX | Sortida TTL connectada a una entrada RX externa. |
| 5 | AUX | Sortida per despertar un microcontrolador . |
| 6 | VCC | Entrada d'alimentació 5V. |
| 7 | GND | Entrada de GND, 0V. |

2.1.5.2 Modes de treball del mòdul AS32-TTL-100:

| Mode | M1 | M0 | Estat | Observacions |
|---------------------------------|----|----|---|--|
| Mode 0 Normal. | 0 | 0 | Canals UART i Wireless oberts. Transmissió transparent activada. | El Receptor ha de treballar en mode 0 o 1. |
| Mode 1 Desperta. | 1 | 0 | Canals UART i Wireless oberts però ara s'afegirà un codi de preàmbul abans de cada transmissió de dades de tal forma que pot despertar el receptor que haurà de treballar en mode 2. | El receptor pot treballar en mode 0, 1 o 2. |
| Mode 2 estalvi d'energia. | 0 | 1 | UART desactivada. El mòdul Wireless treballa en mode WOR(Wake On Radio). quan rebi el codi de preàmbul obrirà l'UART per rebre les dades. En aquest mode no es possible la transmissió. | Es obligatori que el transmissor treballi en mode 1. |
| Mode 3 Configuració. | 1 | 1 | Mode per configurar el mòdul. | |

Com es pot veure en la figura 13, en aquest projecte s'utilitza el mode 0 per defecte.

Utilitzar aquest mode implica que per la Transmissió de dades, quan la longitud de dades que introdueix l'usuari arriba 58 Bytes, el mòdul iniciarà la transmissió sense fils. En aquest moment, l'usuari pot continuar ingressant dades a transmetre. quan l'usuari introdueix menys de 58 Bytes, el mòdul espera el temps de 3 Bytes, si no segueix introduint dades es considerarà acabat, i el mòdul enviara tots els paquets de dades a traves de la connexió sense fils.

El mòdul sempre activa la funció de recepció inalàmbrica, i pot rebre paquets de dades des del mode 0 o el mode 1. Després de rebre el paquet de dades, la sortida AUX passa a nivell baix i amb un retràs de 2ms aquestes dades passen al port serie pel seu pin TX. En aquest punt les dades passen del pin TX al microcontrolador.

2.1.6 Mòdul GPS GY-NEO6MV2.

El mòdul GPS GY-NEO6MV2 és un receptor GPS autònom que té el motor de posicionament u-blox 6 d'alt rendiment. Aquests mòduls GPS poden ser connectats fàcilment a un microcontrolador a partir del bus SPP (Serial Port Protocol).

La família de receptors GPS NEO-6 estan dissenyats per tenir una mida petita, petit cost, i petit consum.

La tensió d'alimentació de la família NEO-6 va des de 1,75V a 3,6V tot i això molts dels mòduls incorporen un regulador de voltatge de tal forma que també és possible alimentar-lo amb 5 V procedents del microcontrolador.

Segons la documentació, els mòduls NEO-6 tenen un temps d'arrancada de uns 30s en fred i 1s en calent i la seva freqüència màxima de mesura és de 5Hz. Tenint una precisió de 2,5m, amb velocitat de 0,1m/s i en orientació de 0,5°, uns valors molt acceptables per un sistema de posicionament GPS. Posteriorment a l'apartat de les proves 2.3.9 podrem comprovar si aquests valors es troben dins la realitat.

Aquests aparells són molt utilitzats per projectes d'electrònica de mides reduïdes on sigui necessari detectar el posicionament d'un objecte en el planeta Terra.

En aquest projecte volem utilitzar aquest mòdul per poder detectar la localització de l'aparell quan aparegui una situació d'emergència.

Característiques del dispositiu:

- Alimentació: 3,3 V – 5 V.
- Consum: 70 mA.
- Guany de l'antena: 50 dB.

- Temperatura d'operació: -40 a 85 °C.
- Interfícies de comunicació: UART.
- Sensibilitat: -160 dBm.

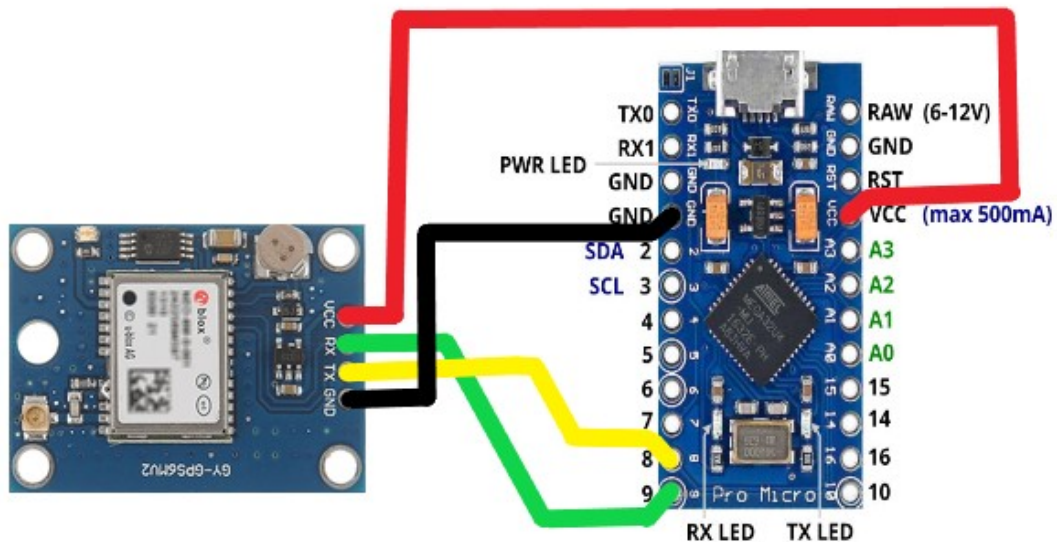


Figura 15: Cablejat del mòdul GPS GY-NEO6MV2 al microcontrolador bus Serial

En la figura 14 es pot apreciar les connexions fetes entre el mòdul GPS i el microcontrolador, aquest és alimentat mitjançant els 5V procedents del microcontrolador a partir del seu pin VCC. També es pot apreciar que el mòdul GPS incorpora comunicació UART, i aquesta no està connectada als pins TX i RX del microcontrolador, ja que aquests estan ocupats per la comunicació amb altres mòduls en el nostre projecte, en comptes d'utilitzar aquests pins, en aquesta ocasió, s'utilitzen els pins 8 i 9 del microcontrolador recolzant-nos amb la llibreria SoftwareSerial per tal de poder habilitar un segon port Serial, en aquest cas digital en el microcontrolador i poder-nos comunicar amb el mòdul GPS.

2.1.7 Disseny del circuit imprès.

Un cop muntats i provats tots els components en la placa de prototipat, es dissenya un circuit imprès a partir del programa KiCad. Aquest programa de software lliure és utilitzat per la automatització del disseny electrònic. Aquest facilita el disseny d'esquemes per circuits electrònics i la seva conversió a placa de circuit imprès.

Dins aquest programa podem trobar:

- KiCad: L'administrador d'arxius.
- Eschema: L'editor d'esquemes.
- Pcb Layout: L'entorn de disseny de circuits impresos (PCB).

- Gerbvew: Visualitzador d'arxius Gerber.
- Bitmap2Component: Eina que converteix imatges a petjades de components (footprints) per realitzar «PCB artwork».

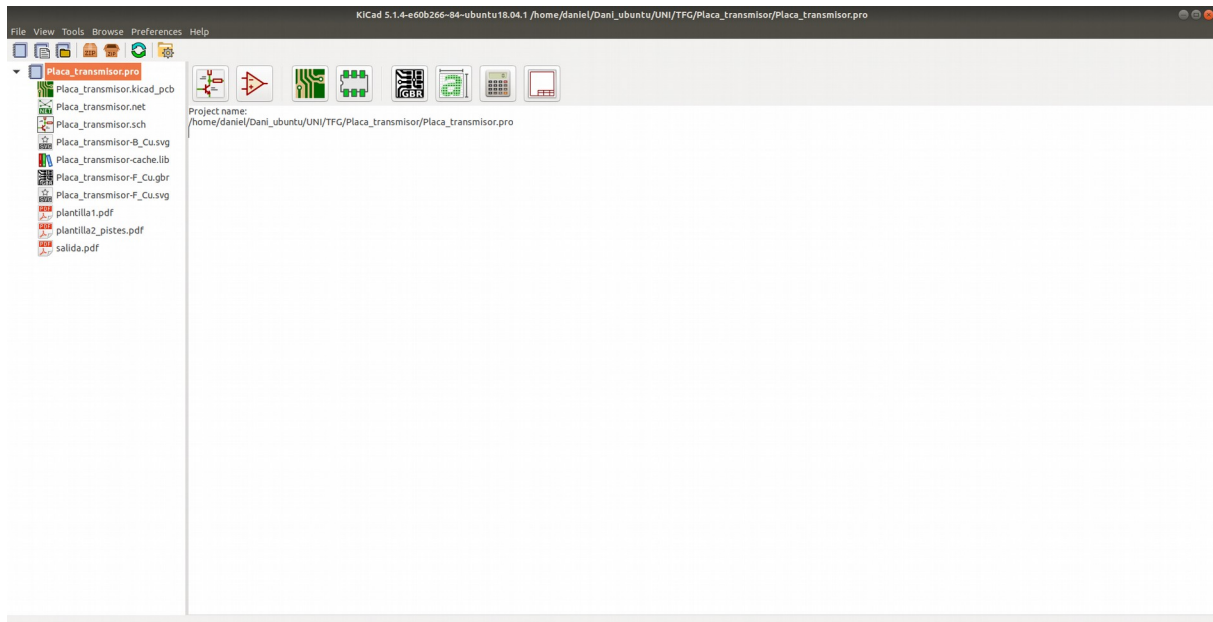


Figura 16: Programa KiCad

En la figura 15, es pot apreciar l'entorn de desenvolupament de la placa de circuit imprès a partir del programa KiCad. En Aquesta figura, es pot apreciar la pantalla principal del programa des de la qual es gestiona el projecte. Per començar, es crea un esquema elèctric de la placa a partir de l'editor d'esquemes del programa. Posteriorment, s'ha de crear cada una de les petjades (footprints) dels components de la placa per tal de vincular cada una de les petjades amb cada un dels components de l'esquema. Seguidament, es crea una llista de nodes (netlist) a partir de l'editor d'esquemes, per tal de que els programes de Cad puguin fer una relació de quins pins de cada component pertanyen al mateix node o pista. Per finalitzar a partir de l'editor de cad Pcb Layout es dibuixa les pistes de la placa de circuit imprès tot seguint les línies de node generats per el netlist.

2.1.7.1 L'esquema elèctric.

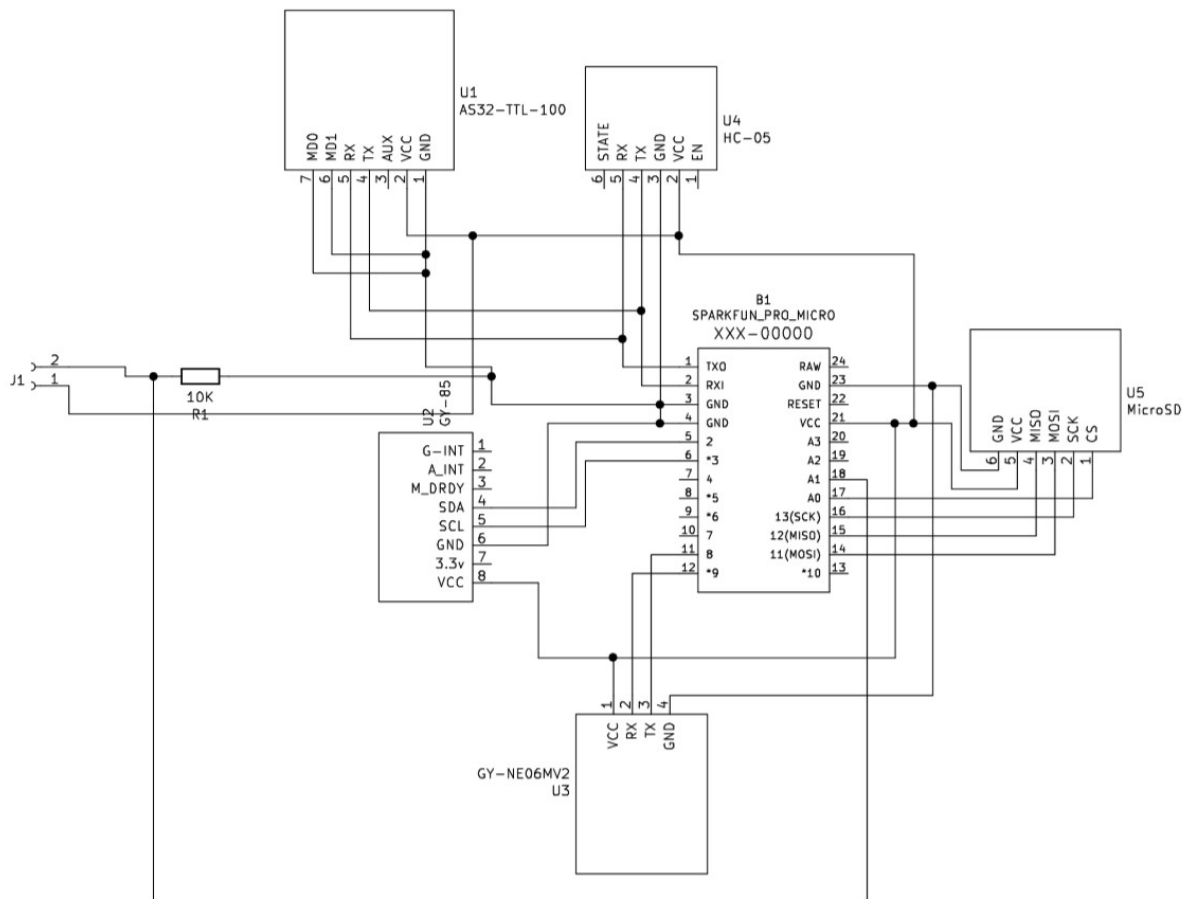

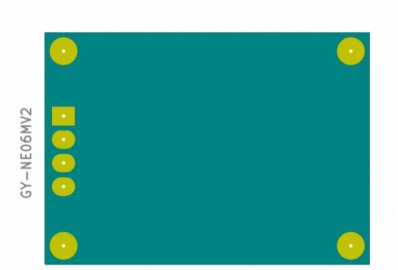




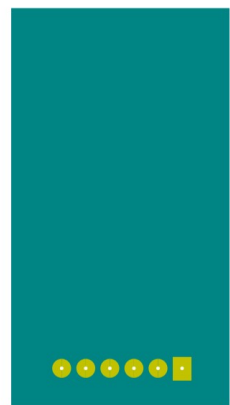
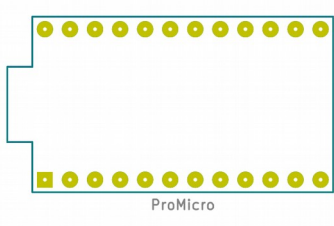
Figura 17: Esquema elèctric de la placa de circuit imprès.

En la figura 16, es pot apreciar l'esquema elèctric de la placa del circuit imprès dissenyat. En aquesta figura, es veuen totes les connexions de les diferents línies de transmissió de la placa, així com el detall de cada una de les potes dels components. És important fixar-nos en les connexions des del microcontrolador Sparkfun_PRO_MICRO que apareix en la figura, ja que aquest és l'element central del projecte i a on estan connectats tots els altres mòduls. Ens podem fixar en el detall que el bus Serial SPP format per els pins 1 i 2 del microcontrolador Sparkfun_PRO_MICRO, es troba compartit entre el mòdul Bluetooth HC-05 i el mòdul LoRa As32-TTL-100. Per tant per aquests dos dispositius extraurem la mateixa informació. També podem veure el detall de totes les alimentacions dels mòduls perifèrics al microcontrolador, aquesta depèn en tots ells del pin 21 VCC del microcontrolador. A partir d'aquí, es pot veure en aquest esquema les connexions dels busus dels altres mòduls. El bus I2C amb les potes 5 i 6 denominades en aquest cas com a 2 i 3 en la placa del microcontrolador i connectades al mòdul IMU Gy-85. El bus SPI pel mòdul de memòria micro SD amb les potes 14, 15, 16 i 17, denominades en aquest cas com a 11, 12, 13 i A0 respectivament en el nostre microcontrolador. I per últim el port Sèrie virtual format per les

potes, 11 i 12, denominades en aquest cas com a 8 i 9 en el microcontrolador connectades al mòdul GPS.

2.1.7.2 Petjades dels components.

| Dispositiu | Petjada del component (footprint). |
|---|---|
| Mòdul de transmissió de dades LoRa AS-32-TTL-100. |  |
| Mòdul GPS GY-NEO6MV2 |  |
| Mòdul de transmissió Bluetooth HC-05. (Vertical) |  |

| | |
|----------------------------|--|
| IMU GY-85 |  |
| Mòdul de memòria micro SD |  |
| Microcontrolador ProMicro. |  |

En la taula anterior, es pot veure la petjada de cada un dels components. Aquesta petjada ha sigut creada a partir de l'editor de petjades, ja que aquestes no estaven incloses a cap llibreria amb l'excepció del microcontrolador, aquesta petjada s'hapogut importar des de la llibreria de Sparkfun. Per la creació de la resta de les petjades, s'ha mesurat cada un dels components amb un instrument de mesura com pot ser un regle o un peu de rei, i s'ha dibuixat. Per fer més fàcil la feina, com que cada component té un pas estàndard entre les seves potes de 2,54 mm s'ha agafat com a model una regleta del pas 2,54 mm amb el nombre de potes que té cada component i s'ha editat el nou component sobre d'aquesta. Cal dir que en aquesta feina és important dibuixar el contorn de cada un dels components, ja que sinó es pot patir problemes en el procés de muntatge i soldadura d'aquests. En aquest projecte tots els components que hem seleccionat per muntar són amb forat. Com que s'ha

hagut de dibuixar tots els components, també s'ha aprofitat per editar cada un dels pads i fer-lo una mica més gran per tal de no tenir problemes a l'hora del muntatge.

2.1.7.3 Traçat de la placa PCB.

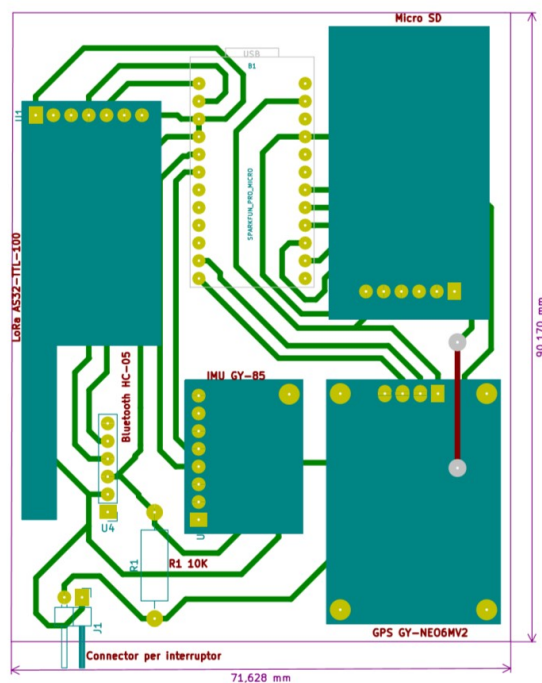


Figura 18: Placa PCB amb els components.

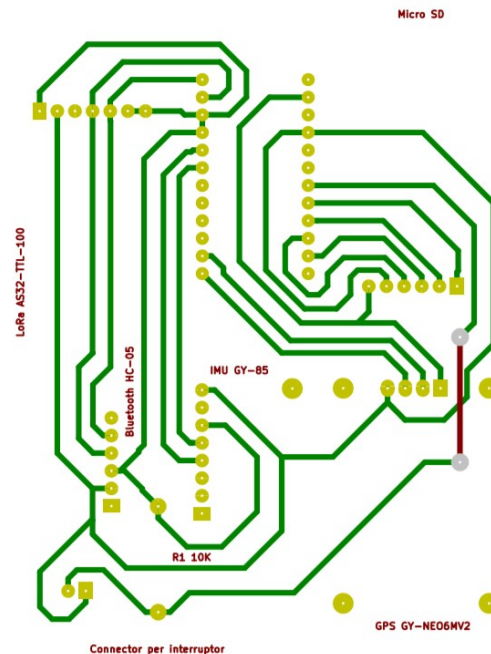


Figura 19: Placa PCB amb dues cares.

En la figura 18, es pot veure el traçat de les pistes en la placa PCB. Aquest traçat, s'ha fet a mà per cada un dels components. En la figura 17, es pot veure la localització dels components en la placa, aquesta localització s'ha modificat a mà mentre s'estava realitzant el traçat de la placa per intentar que les pistes no es creuin, d'aquesta forma podem utilitzar el mínim nombre de capes possibles en el disseny. Tal i com es pot veure en la figura 18, el traçat s'ha efectuat utilitzant només dues capes, dues vies i un pont. Aquest pont s'efectua a partir de un cable rígid correctament aïllat, d'aquesta forma evitarem haver de fabricar la placa a doble cara i evitarem costos.

Per fer el traçat de la placa s'han tingut en compte els següents paràmetres:

- Utilitzar angles obtusos, més de 90° en la placa.
- Tindre un espai suficient entre les pistes llargues que siguin paral·leles.
- Dimensions de les pistes de 0.8 mm.

- Dimensions dels pads 2 mm.

Utilitzar aquestes propietats per el disseny de la placa ens assegura:

- Evitar antenes no desitjades en la placa.
- Evitar punts calents en la placa per un accés d'intensitat en el punt.
- Evitar capacitats paràsites entre les pistes.
- Tindre una resistència de pista baixa.
- Poder foradar i soldar cada pad de forma correcte.

Utilitzar tots aquests criteris en el disseny, hem aconseguit unes dimensions aproximades de la placa de 9 cm de llargada i 7 cm d'amplada, com es pot apreciar en la figura 17.

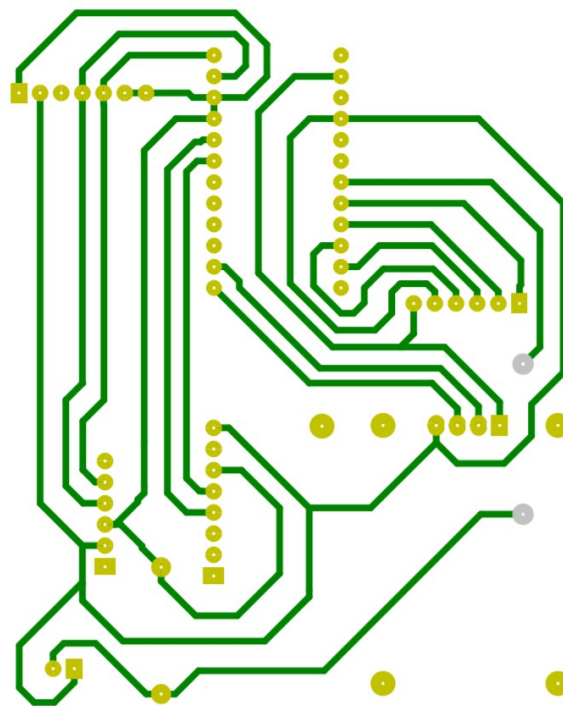


Figura 20: Placa PCB cara de la Impressió.

En la figura 19, es pot veure el disseny de la placa PCB preparat per la seva fabricació. En aquesta figura, s'han omès totes les petjades dels components, per tal de poder transportar la totalitat de la imatge a la placa PCB al coure, si les petjades encara estiguessin visibles com en la figura 17, o hi hagues l'esquema de la figura 18, tindriem pistes creuades i curtcircuits en la placa i aquesta no funcionaria correctament. Així doncs, a partir d'aquesta imatge, farem el fotolit per la fabricació de la placa a partir del mètode químic.

2.1.7.4 Fabricació de la placa PCB.

Per la fabricació de la placa PCB s'utilitza el mètode químic. En concret en aquest projecte s'ha portat a fabricar la placa a una empresa de Puig-reig amb la qual tenim una bona relació i podem seguir el procés de fabricació de la placa amb ells.

Utilitzarem un paquet dissenyat per Ariston per la fabricació de plaques.

Per fer la fabricació de la placa seguim els següents passos:

- (1) Fabricació del fotolit. Per poder fabricar la placa PCB amb el mètode químic, primerament s'ha d'imprimir el disseny que volem transportar a la placa en un fotolit per anar bé a partir d'una impressora laser. El fotolit, és un paper plàstic totalment transparent en el qual es pot imprimir. Abans de fer la impressió i a partir de les propietats que ens dona el software de la impressora, posarem la màxima resolució possible en la impressora, i seleccionarem el mode de transparència o fotolit.

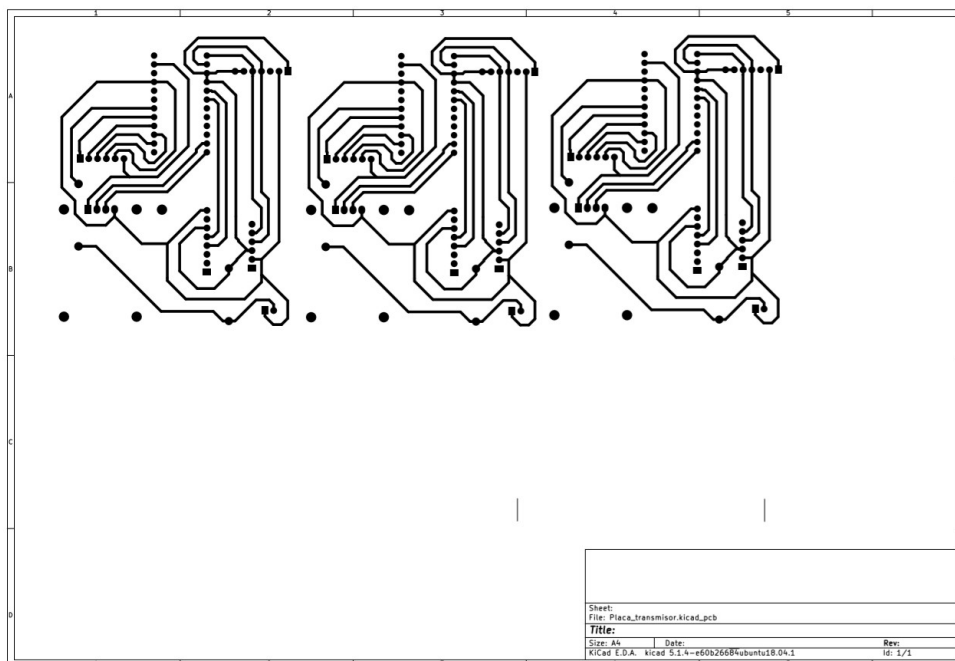


Figura 21: Fotolit per impressió

- (2) Insolació. Un cop imprès el fotolit, s'agafa una placa verge d'una capa de coure, traurem el plàstic protector i seguidament el posarem en una màquina insoladora. Aquesta màquina aplicarà llum infraroja a la placa de coure i debilitarà l'estructura de la capa fotosensible que no estigui protegida per la impressió del fotolit.

- (3) Rebel·lat. Un cop la placa ja ha estat irradiada aproximadament 10 minuts. Agafem una gaveta i posem sosa càustica o Hidròxid de Sodi, NaOH. Farem una dissolució amb aigua, aplicarem una relació de 16 grams de sosa càustica per cada litre d'aigua H_2O . En les figures 21, 22, 23 i 24, podem apreciar el procés de rebel·lat. En la figura 21, veiem la gaveta amb la dissolució de NaOH efectuada. En la figura 22, veiem la placa del circuit imprès submergida en el líquid revelador. Els efectes d'aquest líquid un cop transcorreguts uns minuts es pot veure en la figura 23, és important anar movent la gaveta per que el líquid faci efecte ja que va desfent la capa protectora mica en mica. Un cop la capa fotosensible ha desaparegut, ja podem extreure la placa del líquid revelador, aplicant una mica d'aigua acabem de netejar la placa, podem veure el resultat a la figura 24.



Figura 22: Dissolució de NaOH Sosa Càustica.



Figura 23: Immersió de la placa en el rebel·lat.



Figura 25: Efecte del rebel·lat

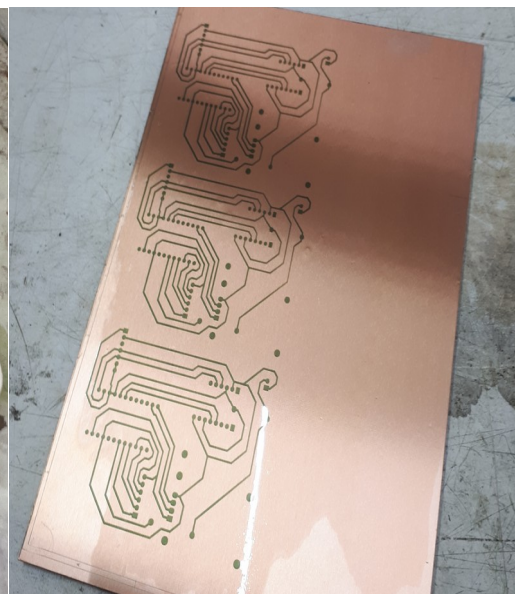


Figura 24: Rebel·lat finalitzat.

- (4) Atacat del coure. Un cop ja tenim la placa rebel·lada, Procedim a eliminar el coure sobrant de la placa, el qual no està protegit per la capa fotosensible. En la figura 25 podem veure el preparat que ven Ariston per atacar el Coure. En concret aquest està compost per una ampolla de Acid Clorhídric HCl en format líquid. I un pot de Perborat sòdic NaBO_3 , en aquest cas en format solid, petits cristalls. Per fer la solució d'atacat pel coure, es fa una dissolució del Perborat sòdic en l'àcid clorhídric en una relació de 1 a 1. Si utilitzem un quart d'ampolla d'àcid clorhídric, a aquest li posarem un quart d'ampolla de Perborat sòdic, això és el que ens recomana Ariston. S'ha d'anar en compte, ja que la reacció és una mica exotèrmica.



Figura 26: Preparat Ariston per atacar el coure.



Figura 27: Placa de circuit dins la solució àcida.

Un cop feta la solució en una gaveta, submergirem la placa de coure ja rebel·lada, tal i com es pot veure en la figura 26. Un cop inserida la placa, s'ha d'anar movent la solució per tal de treure el coure ja reaccionat de sobre la placa. Veurem que mentre es vagi efectuant la reacció, la solució va modificant el color cada cop a un verd més fosc.

quan l'àcid ja s'hagi menjat tot el coure no desitjat, ràpidament extraurem la placa de la gaveta per aturar la reacció i la rentarem amb aigua per treure tot rastre d'àcid. Un cop aturada la reacció, tindrem el circuit imprès amb les pistes de coure encara protegida per la capa fotosensible tal i com es pot veure en la figura 27. Per retirar aquesta capa fotosensible, podem utilitzar 2 mètodes, el més utilitzat és una mica de alcohol amb un paper, el segon mètode que podem utilitzar és el polit de la placa amb paper de vidre, però amb aquest procés també rebaixarem una mica de material de coure de les pistes i per tant no és tant recomanat.

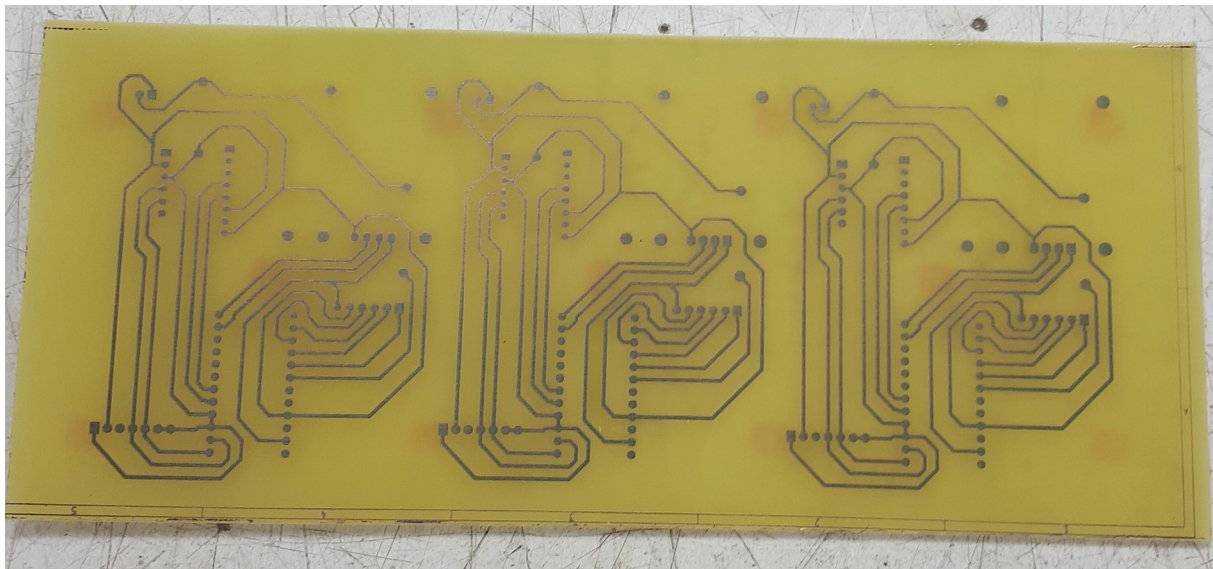


Figura 28: Placa finalitzada.

Finalment només cal trepanar els pads de cada una de les plaques utilitzant una broca de 0.8mm per alguns components i la de 1mm per alguns altres. Com ja s'ha mencionat en l'apartat del disseny, els pads han estat sobredimensionats per tal de no tindre problemes en aquesta fase de trepanat per tant podem utilitzar la broca de 1mm sense cap problema en cada un d'ells. Un cop trepanada tota la placa, ja estem en disposició d'inserir i soldar amb estany cada un dels components que la componen.

2.2. Software.

En aquest projecte el software s'ha desenvolupat mica en mica, provant en mode estàtic cada un dels mòdols, mentre aquests encara estaven muntats sobre la placa de prototipat. En aquest apartat s'explica l'evolució que ha petit l'algoritme des de l'inici fins que hem tingut la placa muntada sobre el suport en la motocicleta.

Per fer la feina més fàcil, hem utilitzat llibreries per tal de controlar el hardware seleccionat per el projecte.

2.2.1 Microcontrolador Pro micro.

Per poder desenvolupar el projecte amb l'entorn de desenvolupament d'Arduino, és necessari instal·lar la placa Pro micro de Sparkfun com a placa disponible dins de la IDE d'Arduino. Es fa a partir de la pestanya Preferencias que podem trobar a Archivo → Preferencias.

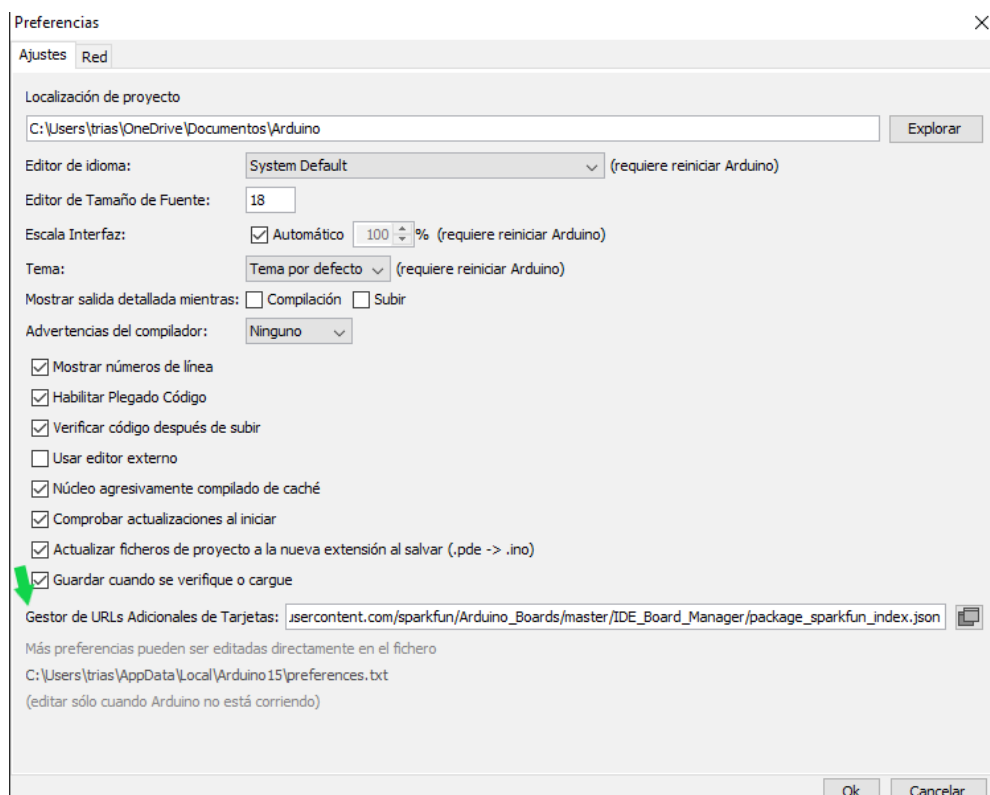


Figura 29: Arduino, Archivo -> Preferencias

En aquest punt copiarem el següent enllaç:

https://raw.githubusercontent.com/sparkfun/Arduino_Boards/master/IDE_Board_Manager/package_sparkfun_index.json

Posteriorment anirem al gestor de targetes ubicat a:

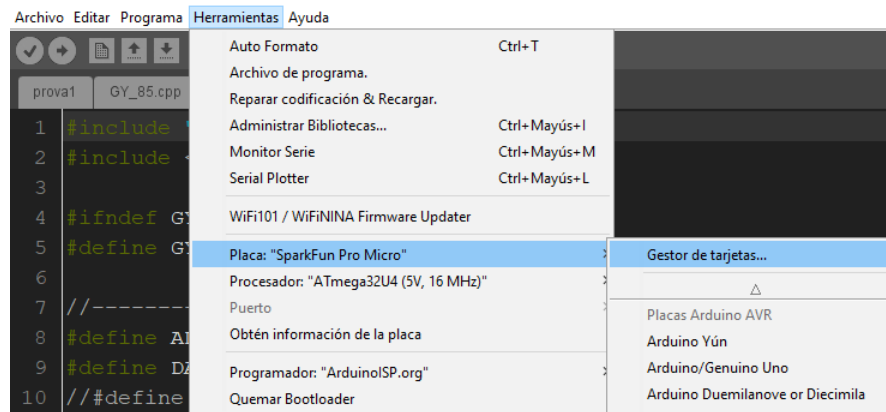


Figura 30: Gestor de targetes

Un cop obert el gestor de targetes, escriurem en ell Sparkfun AVR, i instal·larem el paquet pertinent a les plaques AVR desenvolupades per Sparkfun tal i com es pot veure en la figura 30.

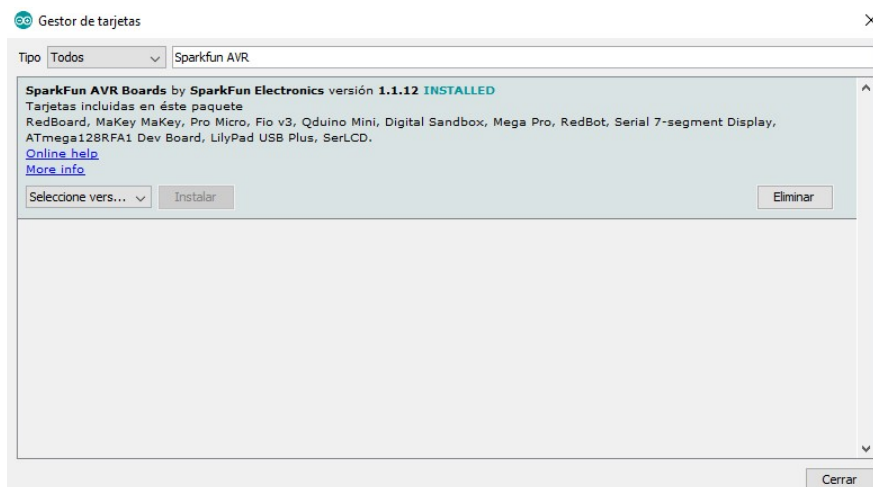


Figura 31: Gestor de targetes, busca de targetes Sparkfun

Un cop instal·lades les targetes Sparkfun ja estarem en disposició de poder seleccionar la placa Pro micro tal i com es veu en la figura 31, així ja podrem carregar el nostre primer programa en ella.

Abans de la carrega del primer programa, haurem de tindre en compte que de la placa hi ha 2 versions la de 3,3V i 8 MHz, i la de 5V i 16 MHz. En el nostre cas escollirem la de ATmega32U4(5V, 16 MHz) tal i com es veu en la figura 32.

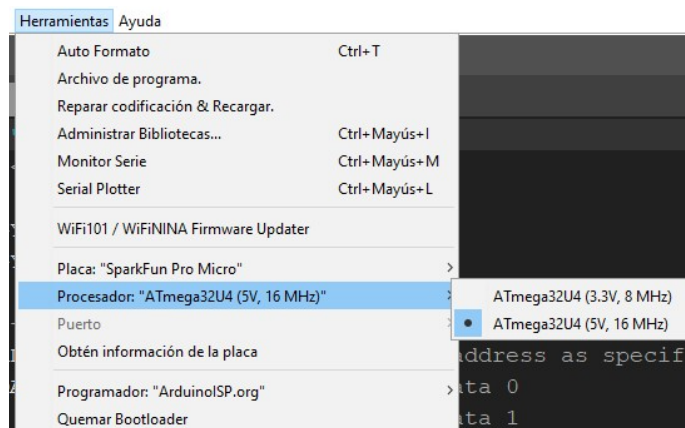


Figura 33: Tipus de placa

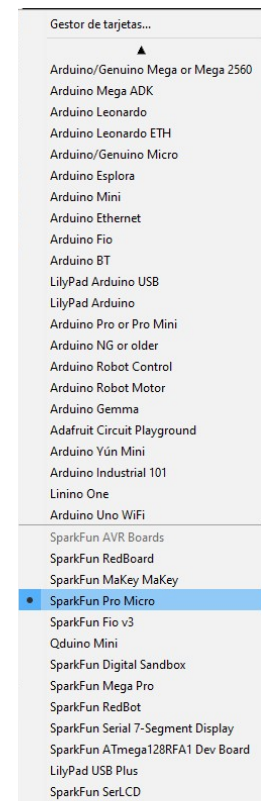


Figura 32:
Selecció de la
placa

2.2.2 Lectura de les dades del sensor GY- 85.

Per fer la feina més fàcil amb el mòdul GY-85 a partir de la comunicació mitjançant la línia de comunicació amb I2C, utilitzem la llibreria Gy-85 extret del següent lloc <https://github.com/sqrtno/GY-85-arduino.git>.

Per fer que el programa funcioni, és necessari la següent estructura de fitxers.



És important que aquests fitxers estiguin dins una carpeta amb nom prova1 si volem que funcioni amb l'entorn de desenvolupament d'Arduino.

Primer de tot anem a veure com actuen els sensors del nostre integrat IMU GY-85.

Per fer-ho utilitzarem l'algorisme següent:

- Programa 1: Lectura del sensor.

```
#include "GY_85.h"

#include <Wire.h>

GY_85 GY85;    //create the object

long tiempo_prev,dt;

void setup()
{
  //inicializa la comunicación serial
  Serial.begin(9600);
  // conf GY-85
  Wire.begin();
  delay(4);
  GY85.init();
  delay(4);
  tiempo_prev=millis();
}

void loop()
{
  static int temps=5;

  // Escalat per 4G.

  float ax = GY85.accelerometer_x( GY85.readFromAccelerometer() ) * 0.082;
  float ay = GY85.accelerometer_y( GY85.readFromAccelerometer() ) * 0.082;
  float az = GY85.accelerometer_z( GY85.readFromAccelerometer() ) * 0.082;
  //int cx = GY85.compass_x( GY85.readFromCompass() );
  // int cy = GY85.compass_y( GY85.readFromCompass() );
  // int cz = GY85.compass_z( GY85.readFromCompass() );

  float gx = GY85.gyro_x( GY85.readGyro() ) / 14.375;
  float gy = GY85.gyro_y( GY85.readGyro() ) / 14.375;
  float gz = GY85.gyro_z( GY85.readGyro() ) / 14.375;
```

```
Serial.print ( ax );  
delayMicroseconds(temps);  
Serial.print ( ' ' );//y  
delayMicroseconds(temps);  
Serial.print ( ay );  
delayMicroseconds(temps);  
delayMicroseconds(temps);  
Serial.print ( ' ' );//z  
delayMicroseconds(temps);  
Serial.print ( az );  
delayMicroseconds(temps);  
Serial.print ( ' ' );  
delayMicroseconds(temps);  
Serial.print ( gx );  
delayMicroseconds(temps);  
Serial.print ( ' ' );  
delayMicroseconds(temps);  
Serial.print ( gy );  
delayMicroseconds(temps);  
Serial.print ( ' ' );  
delayMicroseconds(temps);  
Serial.print ( gz );  
delayMicroseconds(temps);  
Serial.print ("\n");  
delayMicroseconds(temps);  
  
}
```

Observant el programa 1 podem apreciar varis sectors d'aquest programa:

- **Capçalera:** Primerament podem observar les incusions de llibreries necessàries per poder utilitzar certes funcions en el nostre codi. En segon lloc, podem veure la creació de l'objecte GY85 i les veritables globals.
- **Configuracions:** Seguidament, veiem la funció setup. El nostre programa executara aquesta secció en primer lloc, i només passarà per aquesta secció una vegada. En aquesta funció, s'inicialitza el port serie a partir de la funció Serial.begin(9600), el port I2C a partir de la llibreria Wire.begin(), els sensors del nostre IMU Gy-85 a partir de la funció GY85.init(), i finalment capturem el temps en el qual estem sortint d'aquesta funció amb la funció millis().
- **Cos del programa:** En el programa principal, podem apreciar com es van executant les diferents funcions i consultes en els sensors.

Primerament, veiem la creació de les variables en aquest cas les fem estàtiques per que conservin el seu valor entre volta i volta del nostre programa.

Seguidament llegim els sensors, comencem per l'acceleròmetre, i seguim amb el giroscopi. Com es pot veure hi ha 3 lectures per cada un d'ells:

Lectura dels sensors:

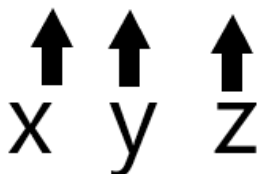
| Sensor | Codi |
|---------------|---|
| Acceleròmetre | <pre>float ax = GY85.accelerometer_x(GY85.readFromAccelerometer())*0.082; float ay = GY85.accelerometer_y(GY85.readFromAccelerometer())*0.082; float az = GY85.accelerometer_z(GY85.readFromAccelerometer())*0.082;</pre> |
| Giroscopi | <pre>float gx = GY85.gyro_x(GY85.readGyro()); float gy = GY85.gyro_y(GY85.readGyro()); float gz = GY85.gyro_z(GY85.readGyro());</pre> |

En aquest cas s'ha escalat les dades en els dos sensors, d'aquesta forma les dades són intel·ligibles per nosaltres de forma contraria aquestes només són números els quals no els podem entendre correctament.

- **Escalat dels sensors:** Per fer les dades dels sensors tractables, s'escalen els valors de la lectura. Per poder-ho fer seguirem la següent metodologia:

Acceleròmetre: Per poder fer l'escalat en l'acceleròmetre, farem la lectura del sensor sense escalar, i escriurem el valor en el port serie. Cal dir que els valors visibles en el port serie variaran segons la posició de l'acceleròmetre i el seu escalat intern, aquest pot ser de 2, 4, 8 o 16 G. En la prova efectuada, el sensor el tenim amb un escalat de 4 G. I el posem pla per l'eix de les Z. El retorn del port serie és el següent:

```
0.00 0.00 118.00
0.00 0.00 119.00
0.00 0.00 119.00
0.00 0.00 119.00
0.00 0.00 119.00
```



A partir de la figura 33 podem veure les acceleracions sense escalar que ens retorna el sensor estant configurat a 4 G. Com que veiem 0 en l'eix X i 0 en l'eix Y, podem deduir que l'acceleròmetre està totalment pla i que la força de la gravetat està actuant totalment sobre l'eix de les Z. Sabent que la força de la gravetat és 9.81 m/s^2 . Podem establir la següent equació d'escalat:

$$\text{Escalat} = \frac{9,81}{\text{Acceleració en pla}} = \frac{9,81}{119} = 0,082436$$

Figura 34: Acceleracions sense escalat 4G

Així aplicant el nostre factor d'escalat, en la lectura de l'acceleròmetre podem aconseguir els resultats en m/s^2 , amb els quals ja podem operar.

Giroscopi: El giroscopi és un sensor que llegeix revolucions per segon, tot i que no entrega les dades en aquesta magnitud, per poder llegir les rotacions en $^\circ/\text{s}$ s'ha d'escalar la mesura a partir sensibilitat del factor d'escala. Aquesta informació la podem trobar en el datasheet, en la pagina de característiques elèctriques i especificacions del sensor. Podem veure la línia marcada en la figura 34 on es mostra el valor d'escalat $14,375 \text{ LSB}(^\circ/\text{s})$ el qual utilitzarem per dividir el valor llegit del sensor. Cal dir que la llibreria GY-85 ja incorpora aquest escalat, per tant per extreure l'angle la rotació en $^\circ/\text{s}$ d'aquest sensor ens limitarem a fer una consulta a la llibreria a partir de la funció donada.

3 Electrical Characteristics

3.1 Sensor Specifications

Typical Operating Circuit of Section 4.2, VDD = 2.5V, VLOGIC = 1.71V to VDD, T_A=25°C.

| Parameter | Conditions | Min | Typical | Max | Unit | Note |
|---|------------------------------|-----|---------|-----|-----------|------|
| GYRO SENSITIVITY | | | | | | |
| Full-Scale Range | FS_SEL=3 | | ±2000 | | °/s | 4 |
| Gyro ADC Word Length | | | 16 | | Bits | 3 |
| Sensitivity Scale Factor | FS_SEL=3 | | 14.375 | | LSB/(°/s) | 3 |
| Sensitivity Scale Factor Tolerance | 25°C | -6 | | +6 | % | 1 |
| Sensitivity Scale Factor Variation Over Temperature | 0°C to 55°C | | ±10 | | % | 2 |
| Nonlinearity | Best fit straight line; 25°C | | 0.2 | | % | 6 |
| Cross-Axis Sensitivity | | | 2 | | % | 6 |
| GYRO ZERO-RATE OUTPUT (ZRO) | | | | | | |
| Initial ZRO Tolerance | | | ±60 | | °/s | 1 |
| ZRO Variation Over Temperature | 0°C to 55°C | | ±40 | | °/s | 2 |
| Power-Supply Sensitivity (1-10Hz) | Sine wave, 100mVpp; VDD=2.2V | | 0.2 | | °/s | 5 |
| Power-Supply Sensitivity (10 - 250Hz) | Sine wave, 100mVpp; VDD=2.2V | | 0.2 | | °/s | 5 |
| Power-Supply Sensitivity (250Hz - 100kHz) | Sine wave, 100mVpp; VDD=2.2V | | 4 | | °/s | 5 |
| Linear Acceleration Sensitivity | Static | | 0.1 | | °/s/g | 6 |

Figura 35: Especificacions del giroscopi per fer l'escalat. (extret de la referència *ITG 3205 datasheet.*)

- **Visualització dels valors:** Per visualitzar els valors utilitzarem la funció `Serial.print(valor)`, el qual ens mostrarà la variable que haguem posat en la zona *valor*. Per poder visualitzar els valors ordenadament, separarem cada un dels valors amb un espai, un cop agüem creat la línia de visualització. En l'últim valor a visualitzar utilitzarem la funció `Serial.println(valor)`, el qual ens farà un salt de línia un cop haguem acabat la impressió del valor.

Els valors que visualitzem són les acceleracions, escalades a m/s^2 , i la rotació que tenim en graus en el moment que nosaltres fem moure el nostre aparell. Això vol dir que quan nosaltres tenim l'aparell quiet. Els valors que podem veure en el giroscopi són 0 per tots els seus eixos. I els de l'acceleròmetre és la acceleració que ens està exercint la força de la gravetat sobre el nostre sensor.

2.2.3 Guardat de les dades en una targeta microSD.

Per poder tractar les dades correctament, utilitzem un mòdul micro SD en el nostre sistema, d'aquesta forma podem guardar les dades en la targeta micro SD i tractar-les posteriorment en un ordinador per poder executar gràfiques i veure que està passant en el sistema en tot moment. Per fer-ho ens ajudarem de la llibreria d'Arduino <SD.h> la qual ens facilitarà molt la gestió i el guardat de les dades en la nostra targeta de memòria. Cal dir que la gestió a partir d'aquesta llibreria farà pujar molt l'ús de memòria de programa per el nostre microcontrolador. Concretament passarem de un 33% que ocupava el programa 1 vist en l'apartat 2.2.2.1, a un 68% que ocuparà en aquest cas.

Per poder escriure la targeta micro SD s'ha efectuat el següent programa:

- **Programa 2: Escriptura de les dades en la targeta micro SD.**

```
#include "GY_85.h"
#include <Wire.h>
#include <SD.h>
#include <string.h>
#define Rad 180/PI

GY_85 GY85;    //create the object
long tiempo_prev,dt;

File Fitxer_dades;

String crea_fitxer(){
    bool fitxer_existent=true;
    int numero_int=0;
    String numero_str=(String) numero_int;
    String Nom_fitxer="Fitxer"+numero_str+".txt";
    while(fitxer_existent){
        if(SD.exists(Nom_fitxer)){
            numero_int++;
            numero_str=(String) numero_int;
            Nom_fitxer="Fitxer"+numero_str+".txt";
        }
        else{
            Fitxer_dades=SD.open(Nom_fitxer, FILE_WRITE); //Creem el fitxer
            numero X
            Fitxer_dades.close();
        }
    }
}
```



```
        fitxer_existent=false;
    }
}

return Nom_fitxer;
}

void setup()
{
    //inicializa la comunicación serial
    Serial.begin(9600);
    // conf GY-85
    Wire.begin();
    delay(4);
    GY85.init();
    delay(4);
    if(!SD.begin(A0)){
        Serial.println("No s'ha pogut inicialitzar");
        return;
    }
    Serial.println("inici ok");
}

void loop()
{
    String Nom_fitxer=crea_fitxer();
    while(true){
        float ax = GY85.accelerometer_x( GY85.readFromAccelerometer() )*0.082;
        float ay = GY85.accelerometer_y( GY85.readFromAccelerometer() )*0.082;
        float az = GY85.accelerometer_z( GY85.readFromAccelerometer() )*0.082;
```

```
float gx = GY85.gyro_x( GY85.readGyro() );
float gy = GY85.gyro_y( GY85.readGyro() );
float gz = GY85.gyro_z( GY85.readGyro() );

dt=(millis()-tiempo_prev)/1000.0;

tiempo_prev=millis();

// escriptura de dades a la SD

Fitxer_dades=SD.open(Nom_fitxer, FILE_WRITE);//obrim el fitxer en mode
lectura

if(Fitxer_dades){

    Fitxer_dades.print(tiempo_prev);

    Fitxer_dades.print(F("\t"));

    Fitxer_dades.print(ax);

    Fitxer_dades.print(F("\t"));

    Fitxer_dades.print(ay);

    Fitxer_dades.print(F("\t"));

    Fitxer_dades.print(az);

    Fitxer_dades.print(F("\t"));

    Fitxer_dades.print(gx);

    Fitxer_dades.print(F("\t"));

    Fitxer_dades.print(gy);

    Fitxer_dades.print(F("\t"));

    Fitxer_dades.println(gz);

    Fitxer_dades.close();//tanquem l'arxiu

}

else{

    Serial.println(F("error al obrir l'arxiu"));

}

}
```

```
}
```

En el programa 2 podem veure les lectures dels sensors del programa 1, però en aquest cas en comptes de visualitzar el valor dels sensors per la pantalla el podem emmagatzemar en la targeta microSD.

Per que les dades no es sobreescriuin entre elles s'ha creat la funció `crea_fitxer()`. Aquesta funció llegeix els fitxers existents en la targeta microSD i mira quin és l'últim fitxer que coincideixi amb el nom `FITXER'nº.txt`. D'aquesta forma l'algoritme agafa el paràmetre 'nº' i l'augmenta en una unitat. Així podem tindre multitud de fitxers ordenats per ordre d'arrancades del sistema.

2.2.4 Detecció dels angles.

- **Acceleròmetre.**

L'acceleròmetre mesura les acceleracions que pateix el nostre sistema. En concret ens interessa mesurar la força de la gravetat, ja que aquesta és coneguda i sabem que és 9.81m/s^2 . En aquest cas el sensor ens permet detectar acceleracions en 3 eixos, X, Y i Z. Així doncs, trigonomètricament podem determinar la orientació del nostre sensor. Per calcular l'angle de inclinació amb l'acceleròmetre utilitzem la fórmula de l'arc tangent.

Tenim 2 formes d'estudiar el nostre sistema. En 2D o en 3D.

- Sistema 2D.

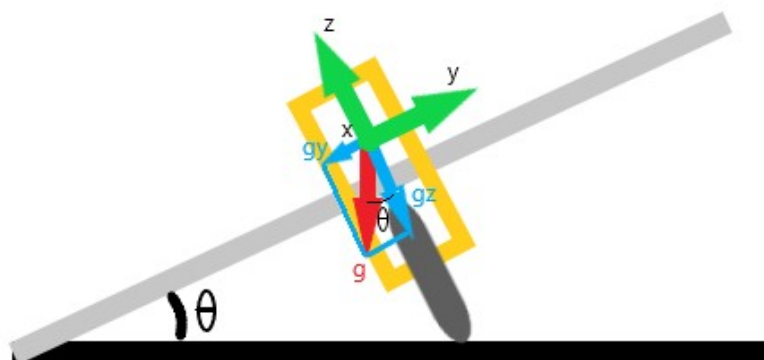


Figura 36: Sistema 2D

Com veiem en la figura 35, podem determinar l'angle a partir de les acceleracions mesurades pel sensor amb els eixos Y i Z. Quan el sistema es troba en posició vertical, l'eix Y i l'eix X són paral·lels al terra, i per tant l'acceleració en l'eix Y i l'eix X és 0, però en aquest cas l'eix Z és perpendicular al terra i per tant l'acceleració en l'eix Z és 9.81 m/s^2 . Quan el sistema està inclinat cap a un dels dos costats l'eix X no modifica el seu estat, segueix sent paral·lel al terra, però l'eix Y deixa de ser-ho modificant la seva component vectorial i incrementant el seu valor. El seu valor serà màxim quan el sistema es trobi a 90° i en aquell moment la seva component vectorial valdrà 9.81 m/s^2 , la qual cosa voldrà dir que es troba perpendicular al terra. En el cas de l'eix Z, la seva evolució serà contrària a l'eix Y. Veiem la màxima component vectorial en l'eix Z quan el sistema es troba a 0° , i veiem la mínima component en aquest eix, quan el sistema es troba a 90° . Així doncs es relativament senzill establir quan el sistema es troba a 0° o a 90° , però no es tan fàcil determinar l'angle quan el sistema es troba entre ells. En aquest cas, la força de la gravetat és la suma vectorial de les acceleracions detectades per l'eix Y i l'eix Z.

Així doncs podem calcular l'angle d'inclinació a partir de la formula:

$$\theta = \arctan\left(\frac{gy}{gz}\right)$$

Cal dir que els eixos poden variar depenent de com posem l'aparell en el sistema a estudiar.

- Sistema 3D.

El sistema 3D té el mateix comportament que el sistema 2D, però molt més complex, ja que en l'eix de les X que en el cas anterior no es tenia en compte per absència de moviment, en aquest cas l'hem de tenir en compte i per tant afegim una nova variable en la nostre equació. Així el sistema a estudiar es converteix en un sistema d'equacions del qual hem d'extreure la inclinació de cada un dels eixos per separat a partir de la següent formula:

$$\theta_x \arctan\left(\frac{gx}{\sqrt{gy^2 + gz^2}}\right)$$

$$\theta_y \arctan\left(\frac{gy}{\sqrt{gx^2 + gz^2}}\right)$$

$$\theta_z \arctan\left(\frac{\sqrt{gx^2 + gy^2}}{gz}\right)$$

- **Giroscopi.**

El giroscopi com ja s'ha explicat anteriorment és un aparell que mesura velocitat angular, això vol dir que ens dona els graus que rota l'aparell en l'eix corresponent per segon. Això vol dir que ens entrega la rotació en el moment que l'aparell s'està movent, però quan aquest s'atura en la posició que sigui, els valors proporcionats per aquest sensor són 0 en tots els seus eixos. Així doncs, si volem saber l'angle d'inclinació a partir d'aquest sensor, el que hem de fer és integrar el senyal en el temps, ja que d'aquesta forma es va efectuant un sumatori dels graus de rotació proporcionats pel sensor.

$$\theta_y = \theta_{y0} + \omega_y \cdot \Delta t$$

S'ha de tindre en compte que utilitzant aquesta forma de detecció d'angles, existeix un error denominat com a DRIFT, el qual apareix quan movem el sistema i el tornem en la posició inicial. quan fem això esperem que al retornar a la posició inicial tornem a mesurar 0°, però això no succeeix, el que veiem és un valor proper a 0°. Aquest error és generat per l'error en la mesura del temps, o del soroll en les mesures del giroscopi. Aquest error per molt petit que sigui, es va acumulant en el temps fins a esdevenir un error el qual podem considerar important.

Per disminuir el DRIFT existeixen varies metodologies, aquestes es basen majoritàriament en l'aplicació de filtres per disminuir el soroll o amb l'aplicació d'altres sensors i amb els angles calculats per aquests millorar el càlcul del giroscopi.

Un dels millors filtres per eliminar el DRIFT, és el filtre de Kalman, però per la aplicació d'aquest filtre és necessària una bona capacitat de processament computacional, fent molt difícil la seva implementació en el nostre microcontrolador.

- **Implementació d'un filtre Complementari: Acceleròmetre + Giroscopi.**

El filtre Complementari és una bona solució per l'eliminació del DRIFT generat pel giroscopi. Aquest es basa en calcular l'angle a partir de l'acceleròmetre i del giroscopi per separat, i efectuar una suma ponderada de cada un d'ells.

La necessitat de combinar ambdues lectures és que si només utilitzem l'acceleròmetre, aquest és susceptible a les acceleracions producte del moviment del sistema o de forces externes, però en temps llargs l'angle no acumula errors. En

canvi, si s'utilitza únicament el giroscopi, aquest és susceptible a l'acumulació d'errors en temps llargs i aquest ens serveix només per mesures en temps curts.

L'equació utilitzada és la següent.

$$\theta_{\text{FiltreComplementari}} = 0,98 \cdot (\theta_{\text{giroscopi}}) + 0,02 (\theta_{\text{acceleròmetre}})$$

D'aquesta forma l'angle de l'acceleròmetre està passant per un filtre passa baixos, esmorteint les variacions brusques de acceleracions no desitjades, i l'angle calculat per el giroscopi té un filtre passa alts tenint una gran influència quan hi ha rotacions ràpides. Aquest filtre també és aplicable amb altres valors, el 0.98 i 0.02 són un exemple l'única premissa que han de complir és que han de complir aquests valors és que han de sumar 1.

- **Programa 3: Detecció dels angles i implementació del filtre complementari.**

```
#include "GY_85.h"
#include <Wire.h>
#include <SD.h>
#include <string.h>
#define Rad 180/PI

GY_85 GY85;      //create the object

File Fitxer_dades;

String crea_fitxer(){
    bool fitxer_existent=true;
    int numero_int=0;
    String numero_str=(String) numero_int;
    String Nom_fitxer="Fitxer"+numero_str+".txt";
```

```
while(fitxer_existent){
    if(SD.exists(Nom_fitxer)){
        numero_int++;
        numero_str=(String) numero_int;
        Nom_fitxer="Fitxer"+numero_str+".txt";
    }
    else{
        Fitxer_dades=SD.open(Nom_fitxer, FILE_WRITE);//Creem el fitxer
        numero X
        Fitxer_dades.close();
        fitxer_existent=false;
    }
}
return Nom_fitxer;
}

void setup()
{
    //inicializa la comunicaci3n serial
    Serial.begin(9600);
    // conf GY-85
    Wire.begin();
    delay(4);
    GY85.init();
    delay(4);
    if(!SD.begin(A0)){
        Serial.println("No s'ha pogut inicialitzar");
        return;
    }
}
```

```
    Serial.println("inici ok");

}

void loop()
{
    long tiempo_prev;

    float dt;

    float giro_y=0;

    float ax, ay, az;

    float gx, gy, gz;

    float ax_0, ay_0, az_0=0;

    bool primera_volta=false;

    float accel_ang_y_0=0;

    float ang_y, ang_y_prev=0;

    String Nom_fitxer=crea_fitxer();

    tiempo_prev=millis();

    while(true){

        //posada a zero a l'arrencada

        if(primeravolta==false){

            ax_0=GY85.accelerometer_x(GY85.readFromAccelerometer())*0.082;

            ay_0=GY85.accelerometer_y(GY85.readFromAccelerometer())*0.082;

            az_0=GY85.accelerometer_z(GY85.readFromAccelerometer())*0.082;

            accel_ang_y_0=atan(ay_0/az_0)*(Rad);

            primera_volta=true;

        }

        //lectura dels sensors
```



```
ax = GY85.accelerometer_x( GY85.readFromAccelerometer() )*0.082;
ay = GY85.accelerometer_y( GY85.readFromAccelerometer() )*0.082;
az = GY85.accelerometer_z( GY85.readFromAccelerometer() )*0.082;

gx = GY85.gyro_x( GY85.readGyro() );
gy = GY85.gyro_y( GY85.readGyro() );
gz = GY85.gyro_z( GY85.readGyro() );

dt=(millis()-tiempo_prev)/1000.0;

tiempo_prev=millis();

//Calcul de l'angle amb l'acceleròmetre
float accel_ang_y=atan(ay/az)*(Rad)-accel_ang_y_0;

//calcul angle amb giroscopi
giro_y=ang_y_prev+(gy)*dt;

//filtre complementari
ang_y = 0.98*(giro_y) + 0.02*accel_ang_y;

// escriptura de dades a la SD

Fitxer_dades=SD.open(Nom_fitxer, FILE_WRITE);//obrim el fitxer en mode
lectura

if(Fitxer_dades){

    Fitxer_dades.print(tiempo_prev);

    Fitxer_dades.print(F("\t"));

    Fitxer_dades.print(ay);

    Fitxer_dades.print(F("\t"));

    Fitxer_dades.print(az);

    Fitxer_dades.print(F("\t"));

    Fitxer_dades.print(gy);

    Fitxer_dades.print(F("\t"));

    Fitxer_dades.print(giro_y);

    Fitxer_dades.print(F("\t"));
```

```
Fitxer_dades.println(ang_y);  
Fitxer_dades.close();//tanquem l'arxiu  
}  
else{  
    Serial.println(F("error al obrir l'arxiu"));  
}  
ang_y_prev=ang_y;  
}  
}
```

En el programa 3 podem veure els càlculs que extreuen les inclinacions del sistema a partir del valor proporcionat per l'acceleròmetre i el giroscopi. Com es pot veure en l'algoritme s'està utilitzant el model bidimensional, ja que la carrega de càlcul del model tridimensional és molt més feixuga per el nostre microcontrolador.

Un cop calculats els angles mitjançant els sensors, s'aplica el filtre complementari per tal d'extreure l'angle total i aprofitar així el millor de cada un dels dos mètodes. En aquest cas, s'està utilitzant una ponderació del 98% per el giroscopi i un 2% per l'acceleròmetre. Posteriorment en l'apartat de proves podem veure com reacciona el nostre sistema a diferents ponderacions del filtre.

Un cop hem recopilat les dades, aquestes s'emmagatzemen en la targeta SD aprofitant el Programa 2.

2.2.5 Transmissió de les dades amb el mòdul LoRa.

Un cop ja es detecten els angles, es vol donar una aplicació telemàtica al sistema, per tal de poder veure aquests angles de forma inalàmbrica, i en temps real en un abast relativament elevat. Per fer-ho s'utilitza el mòdul LoRa AS32-TTL-100. Aquest mòdul rebrà les dades extretes per el port Serial1 i les transmetrà a una freqüència de 433 MHz per tal que un altre mòdul les descodifiqui i desxifri.

Per fer-ho utilitzem el següent programa.

- Programa 4: Transmissió de dades inalàmbrics LoRa.

```
#include "GY_85.h"
#include <Wire.h>
#include <SD.h>
#include <string.h>
#define Rad 180/PI

GY_85 GY85;    //create the object

int RXLED = 17;
File Fitxer_dades;

String crea_fitxer(){
    bool fitxer_existent=true;
    int numero_int=0;
    String numero_str=(String) numero_int;
    String Nom_fitxer="Fitxer"+numero_str+".txt";

    while(fitxer_existent){
        if(SD.exists(Nom_fitxer)){
            numero_int++;
            numero_str=(String) numero_int;
            Nom_fitxer="Fitxer"+numero_str+".txt";
        }
        else{
            Fitxer_dades=SD.open(Nom_fitxer, FILE_WRITE); //Creem el fitxer
            numero X
            Fitxer_dades.close();
            fitxer_existent=false;
        }
    }
}
```

```
    }  
  }  
  return Nom_fitxer;  
}  
void setup()  
{  
  pinMode(RXLED, OUTPUT);  
  //inicialitza la comunicació serial  
  Serial.begin(9600);  
  //inicialitza la comunicació serial1  
  Serial1.begin(9600);  
  // conf GY-85  
  Wire.begin();  
  delay(4);  
  GY85.init();  
  delay(4);  
  if(!SD.begin(A0)){  
    Serial.println("No s'ha pogut inicialitzar");  
    return;  
  }  
  Serial.println("inici ok");  
}  
void loop()  
{  
  long tiempo_prev;  
  float dt;  
  float giro_y=0;  
  
  float ax, ay, az;  
  float gx, gy, gz;  
  float ax_0, ay_0, az_0=0;  
  bool primera_volta=false;  
  float accel_ang_y_0=0;
```

```

float ang_y, ang_y_prev;
String Nom_fitxer=crea_fitxer();
tiempo_prev=millis();
while(true){
    //posada a zero a l'arrencada
    if(primer_volta==false){
        ax_0=GY85.accelerometer_x(GY85.readFromAccelerometer())*0.082;
        ay_0=GY85.accelerometer_y(GY85.readFromAccelerometer())*0.082;
        az_0=GY85.accelerometer_z(GY85.readFromAccelerometer())*0.082;
        accel_ang_y_0=atan(ay_0/az_0)*(Rad);
        primera_volta=true;
    }

    //lectura dels sensors
    ax = GY85.accelerometer_x( GY85.readFromAccelerometer() )*0.082;
    ay = GY85.accelerometer_y( GY85.readFromAccelerometer() )*0.082;
    az = GY85.accelerometer_z( GY85.readFromAccelerometer() )*0.082;
    gx = GY85.gyro_x( GY85.readGyro() );
    gy = GY85.gyro_y( GY85.readGyro() );
    gz = GY85.gyro_z( GY85.readGyro() );
    dt=(millis()-tiempo_prev)/1000.0;
    tiempo_prev=millis();
    //Calcul de l'angle amb l'acceleròmetre
    accel_ang_y=atan(ay/az)*(Rad)-accel_ang_y_0;
    //calcul angle amb giroscopi
    giro_y=ang_y_prev+(gy)*dt;
    //filtre complementari
    ang_y = 0.98*(giro_y) + 0.02*accel_ang_y;
    Serial1.println(ang_y); //Transmissió inalàmbrica
    // escriptura de dades a la SD
    Fitxer_dades=SD.open(Nom_fitxer, FILE_WRITE);//obrim el fitxer en mode
    lectura
    if(Fitxer_dades){
        Fitxer_dades.print(tiempo_prev);
        Fitxer_dades.print(F("\t"));
        Fitxer_dades.print(ang_y);
    }
}

```

```

Fitxer_dades.print(F("\t"));
Fitxer_dades.print(az);
Fitxer_dades.print(F("\t"));
Fitxer_dades.print(gy);
Fitxer_dades.print(F("\t"));
Fitxer_dades.print(giro_y);
Fitxer_dades.print(F("\t"));
Fitxer_dades.println(ang_y);
Fitxer_dades.close();//tanquem l'arxiu
}
else{
  Serial.println(F("error al obrir l'arxiu"));
}
ang_y_prev=ang_y;
}
}

```

En el programa 4, podem veure com en la funció setup es configura el port Serial1 mitjançant la funció Serial1.begin(9600). Aquest port Serial1, serà utilitzat posteriorment en la funció loop. Aquesta funció s'utilitzarà després de l'aplicació del filtre complementari per tal d'enviar l'angle calculat. Concretament l'enviament d'aquesta dada es portarà a terme per la funció Serial1.println(ang_y).

El que aconseguim amb l'aplicació d'aquest apartat, és la visualització remota de l'angle calculat per el nostre sistema a una distància considerable. Amb l'aplicació que li volem donar en aquest projecte, aquest apartat ens permet veure les inclinacions calculades per el nostre sistema a temps real en una gran majoria de circuits, sempre i quan la normativa de la competició ho permeti.

2.2.6 Transmissió de les dades amb el mòdul Bluetooth.

El que volem aconseguir en aquest apartat, és la transmissió de les dades a curt abast i aprofitar la gran connectivitat que té la tecnologia Bluetooth per tal de incorporar el nostre sistema en el gran món de l'IoT (internet de les coses 'Internet of Things'). A partir del mòdul Bluetooth volem poder connectar el nostre dispositiu en un Smartphone, (telefon intel·ligent) per tal d'adquirir les dades a temps real com

passa amb el mòdul LoRa vist anteriorment, però també per que aquest dispositiu és capaç de connectar-nos a internet i mitjançant l'ús de múltiples aplicacions connectar-nos amb altre gent.

Concretament el que volem és enviar al telèfon un avis de possible caiguda. Aquest avis es donarà quan el sistema superi una certa inclinació aproximada de 65°. Si passa aquesta condició, el sistema entendreà que s'ha patit una caiguda i aquest enviarà un missatge d'alerta al telèfon a partir de la tecnologia Bluetooth.

El sistema Bluetooth Hc-05, és necessari configurar-lo a partir de comandes AT. Aquestes comandes són simples instruccions que introduïrem en el port Serial per tan de comunicar-nos amb ell.

En aquest cas, el mòdul Hc-05 no és del tot fàcil de configurar ja que s'ha de fer entrar en mode configuració expressament. Per fer les configuracions oportunes, aquestes s'hauran de fer externament. El millor en aquest cas és utilitzar un Arduino Uno, i una placa de prototipat per tal de poder fer les connexions necessaries.

Per poder configurar correctament el mòdul, se segueix l'esquema següent:

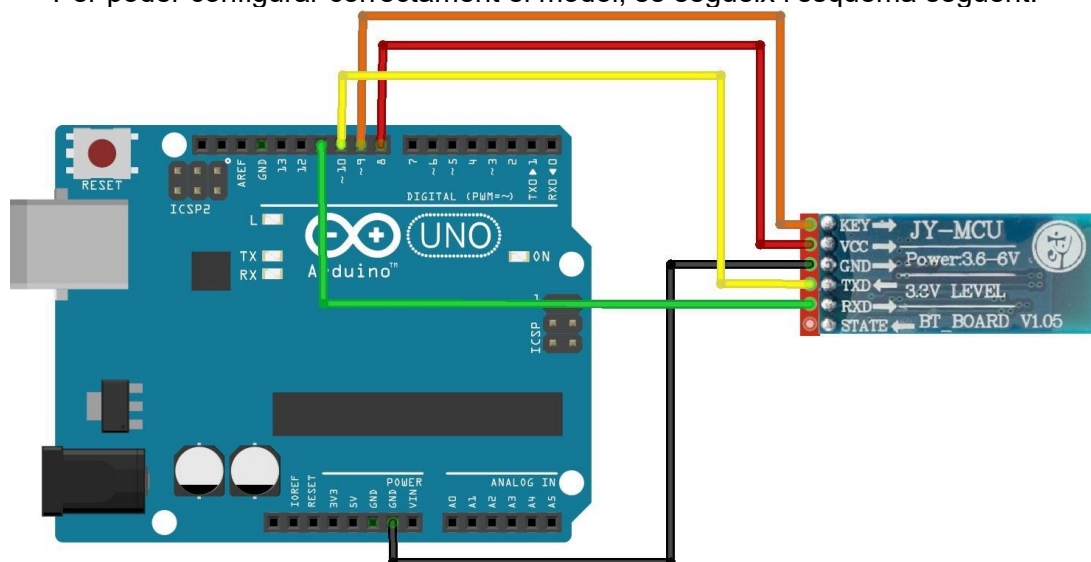


Figura 37: Esquema per la configuració del mòdul Bluetooth.

En la figura 36 podem veure l'esquema que haurem d'implementar en la placa de prototipat per tal de poder configurar el nostre mòdul Bluetooth a partir de les comandes AT.

Posteriorment carregarem el següent programa al microcontrolador Arduino.

- Programa configuració Bluetooth.

```
#include <SoftwareSerial.h>

SoftwareSerial BT1(10, 11); // RX | TX

void setup()
{
  pinMode(8, OUTPUT);          // Alimentació del mòdul Bluetooth
  pinMode(9, OUTPUT);          // Pin per forçar el mode AT
  digitalWrite(9, HIGH);        // Forcem el mode AT al posar una senyal
                                // alta a la pota
  delay (500) ;                 // Espera antes de encender el modulo
  Serial.begin(9600);
  digitalWrite (8, HIGH);       //Enciende el modulo
  Serial.println("Esperant comandes AT:");
  BT1.begin(38400);
}

void loop()
{
  if (BT1.available())
    Serial.write(BT1.read());

  if (Serial.available())
    BT1.write(Serial.read());
}
```

En el programa configuració del mòdul Bluetooth podem veure en la funció setup, com aquest s'ha d'engegar a partir de uns passos determinats en unes de les seves potes. Concretament el que s'ha de fer per posar el mòdul en mode configuració, és

posar una tensió de 5V a la pota Key del mòdul Bluetooth, i posteriorment a això ja el podrem alimentar a partir de la seva pota VCC de la forma que es desitgi, en aquest cas, s'ha optat per posar l'alimentació a la pota 8 de l'Arduino Uno, així tenim més control sobre l'arrancada del mòdul. Podrem distingir que el mòdul ha entrat en mode configuració per que el led auxiliar d'aquest farà pampallugues ràpides durant 1 s aproximadament i posteriorment romandrà aturat durant un altre segon.

Un cop el mòdul ja ha entrat en mode configuració escriurem el següent per la línia de comandes del nostre port Serial:

| Comanda | Resposta | Funció |
|------------------|---------------------|---|
| AT | OK | Si escrivim aquesta comanda i no rebem resposta és possible que no s'hagui inicialitzat correctament el port serial en el nostre mòdul. La velocitat per defecte de fàbrica és 38400 baudis, però es pot haver modificat, o bé per que el mòdul ha estat manipulat prèviament, o per que el fabricant ha modificat aquesta velocitat de transmissió de dades. |
| AT+RESET | OK | Torna el mòdul a la configuració de fàbrica. |
| AT+NAME=Dani | OK, name=Dani | Modifiquem el nom del mòdul Bluetooth. |
| AT+PSWD="1234" | OK, pin= 1234 | Modifiquem la contrasenya del mòdul Bluetooth. |
| AT+UART=9600,0,0 | OK UART=9600,0,0 | Modifiquem la velocitat de transmissió de dades del mòdul Bluetooth. S'ha d'anar en compte en que posterior a aquesta comanda s'ha de tornar a carregar el programa a l'Arduino Uno amb la velocitat del baudrate corresponent. |
| AT+ROLE | OK role = 0 Slave | Configurem el role del nostre mòdul, aquest pot estar com a 0, en tal cas el mòdul es comportarà com un esclau, o en 1, en aquest cas el mòdul es comportarà com a mestre. |

Un cop efectuades les configuracions ja estem preparats per incloure el mòdul Bluetooth en el prototip.

Si ens fixem en l'apartat del hardware, concretament en l'esquema elèctric de la placa, podrem apreciar que tant el mòdul Bluetooth com el mòdul LoRa comparteixen xarxa de comunicacions, per tant, tant un com l'altre emetran la mateixa informació.

En el següent programa podem veure la implementació del mòdul Bluetooth en com a detector de caigudes.

- **Programa 5: Transmissió de dades sense fils LoRa i Bluetooth.**

```
#include "GY_85.h"
#include <Wire.h>
#include <SD.h>
#include <string.h>
#define Rad 180/PI

GY_85 GY85;    //create the object

int RXLED = 17;
File Fitxer_dades;

String crea_fitxer(){
    bool fitxer_existent=true;
    int numero_int=0;
    String numero_str=(String) numero_int;
    String Nom_fitxer="Fitxer"+numero_str+".txt";

    while(fitxer_existent){
        if(SD.exists(Nom_fitxer)){
            numero_int++;
            numero_str=(String) numero_int;
            Nom_fitxer="Fitxer"+numero_str+".txt";
        }
        else{
            Fitxer_dades=SD.open(Nom_fitxer, FILE_WRITE);//Creem el fitxer
            numero X
            Fitxer_dades.close();
        }
    }
}
```

```
        fitxer_existent=false;
    }
}
return Nom_fitxer;
}

void caiguda(){
    //Transmissió de la caiguda.
    Serial1.println("S'ha petit una caiguda!!");
}
void setup()
{
    pinMode(RXLED, OUTPUT);
    //inicialitza la comunicació serial
    Serial.begin(9600);
    //inicialitza la comunicació serial1
    Serial1.begin(9600);
    // conf GY-85
    Wire.begin();
    delay(4);
    GY85.init();
    delay(4);
    if(!SD.begin(A0)){
        Serial.println("No s'ha pogut inicialitzar");
        return;
    }
    Serial.println("inici ok");
}
void loop()
{
    long tiempo_prev;
    float dt;
    float giro_y=0;
```

```
float ax, ay, az;
float gx, gy, gz;
float ax_0, ay_0, az_0=0;
bool primera_volta=false;
float accel_ang_y_0=0;
float ang_y, ang_y_prev;
String Nom_fitxer=crea_fitxer();
tiempo_prev=millis();
while(true){
  //posada a zero a l'arrencada
  if(primeravolta==false){
    ax_0=GY85.accelerometer_x(GY85.readFromAccelerometer())*0.082;
    ay_0=GY85.accelerometer_y(GY85.readFromAccelerometer())*0.082;
    az_0=GY85.accelerometer_z(GY85.readFromAccelerometer())*0.082;
    accel_ang_y_0=atan(ay_0/az_0)*(Rad);
    primera_volta=true;
  }

  //lectura dels sensors
  ax = GY85.accelerometer_x( GY85.readFromAccelerometer() )*0.082;
  ay = GY85.accelerometer_y( GY85.readFromAccelerometer() )*0.082;
  az = GY85.accelerometer_z( GY85.readFromAccelerometer() )*0.082;
  gx = GY85.gyro_x( GY85.readGyro() );
  gy = GY85.gyro_y( GY85.readGyro() );
  gz = GY85.gyro_z( GY85.readGyro() );
  dt=(millis()-tiempo_prev)/1000.0;
  tiempo_prev=millis();
  //Calcul de l'angle amb l'acceleròmetre
  float accel_ang_y=atan(ay/az)*(Rad)-accel_ang_y_0;
  //calcul angle amb giroscopi
  float giro_y=ang_y_prev+(gy)*dt;
  //filtre complementari
  ang_y = 0.98*(giro_y) + 0.02*accel_ang_y;
  Serial1.println(ang_y);//Transmissió inalàmbrica
  // escriptura de dades a la SD
```

```

    Fitxer_dades=SD.open(Nom_fitxer, FILE_WRITE);//obrim el fitxer en mode
lectura
    if(Fitxer_dades){
        Fitxer_dades.print(tiempo_prev);
        Fitxer_dades.print(F("\t"));
        Fitxer_dades.print(ay);
        Fitxer_dades.print(F("\t"));
        Fitxer_dades.print(az);
        Fitxer_dades.print(F("\t"));
        Fitxer_dades.print(gy);
        Fitxer_dades.print(F("\t"));
        Fitxer_dades.print(giro_y);
        Fitxer_dades.print(F("\t"));
        Fitxer_dades.println(ang_y);
        Fitxer_dades.close();//tanquem l'arxiu
    }
    else{
        Serial.println(F("error al obrir l'arxiu"));
    }
    if (ang_y> 65 || ang_y<-65){
        caiguda();
    }
    ang_y_prev=ang_y;
}
}

```

En el programa 5, podem veure com s'ha implementat la funció caiguda a partir d'una simple condició. La condició que tenim en compte és la de tindre un angle superior a 65° o bé un angle inferior a -65°. Si una d'aquestes dues condicions es compleix, el microcontrolador executarà la funció caiguda, la qual enviarà pel port Serial1 la frase "S'ha petit una caiguda!!". Aquesta frase la transmetrà tant el mòdul Bluetooth com el mòdul LoRa, de tal forma que aquesta es podrà descodificar en els dispositius finals corresponents per cada un dels dos sistemes.

2.2.7 El mòdul GPS.

El mòdul Gps va connectat en el microcontrolador a partir de unes potes auxiliars, les quals no pertanyen a cap port de comunicació específicament. Tot i això aquest dispositiu requereix de comunicació Serial a partir de les seves potes TX i RX. Al no tindre disponibles més ports Serial en el microcontrolador s'ha optat per la opció de fer un port Serial software a partir de la llibreria d'Arduino <SoftwareSerial.h>. Aquesta llibreria facilita la feina a l'hora de crear un port serial auxiliar en el dispositiu. Ofereix la possibilitat d'escollir dues potes com a port Serial i poder aprofitar les mateixes funcions que estan disponibles per aquest.

Si ens fixem en l'apartat de Hardware apartat 2.1.7.1 Esquema elèctric, en la Figura 16, podem veure que el mòdul GPS es troba connectat a les potes 8 i 9 del microcontrolador.

Per poder controlar correctament la placa GPS GY-NEO6MV2 a partir del port Serial utilitzem la llibreria <TinyGPS++.h> la qual ens permet adquirir les dades del mòdul GPS de forma senzilla. Aquest mòdul ens pot proporcionar la localització del sistema, la data actual, la hora actual, la velocitat del sistema, el curs del sistema, l'altitud del sistema, la quantitat de satèl·lits els quals està connectat, i la seva precisió horitzontal.

Per utilitzar el mòdul ho farem a partir de les següents funcions:

- **Localització.**

```
if (gps.location.isValid())//LOCALITZACIÓ
{
    Serial1.print("Satelit: ");
    Serial1.println(gps.satellites.value());
    Serial1.print("Localitzacio gps: ");
    Serial1.print(gps.location.lat(),10);//El nombre marca el nombre de
decimals
    Serial1.print(" , ");
    Serial1.println(gps.location.lng(),10);
}
else
{
    Serial1.print("INVALID");
}
```

- **Data.**

```
Serial1.print("  Date/Time: ");
if (gps.date.isValid())//DATA
{
    Serial1.print(gps.date.month());
    Serial1.print("/");
    Serial1.print(gps.date.day());
    Serial1.print("/");
    Serial1.print(gps.date.year());
}
else
{
    Serial1.print("INVALID");
}
```

- **Hora.**

```
Serial1.print(" ");
if (gps.time.isValid())//HORA
{
    if (gps.time.hour() < 10) Serial.print("0");
    Serial1.print(gps.time.hour());
    Serial1.print(F(":"));
    if (gps.time.minute() < 10) Serial.print("0");
    Serial1.print(gps.time.minute());
    Serial1.print(F(":"));
    if (gps.time.second() < 10) Serial.print("0");
    Serial1.print(gps.time.second());
    Serial1.print(F("."));
    if (gps.time.centisecond() < 10) Serial.print("0");
```

```

    Serial1.print(gps.time.centisecond());
}
else
{
    Serial1.print("INVALID");
}

```

En els programes de Localització, data i hora veiem les funcions i l'esquelet del programa que ens permetrà extreure aquests valors del GPS. Aquest GPS té moltes més funcions, per exemple també podríem extreure el rumb i la velocitat de l'objecte, però en aquest cas no les utilitzem ja que el sistema està pensat per alertar en cas d'accident, d'aquesta forma buscarem una activació d'aquestes funcions només en cas d'emergència per estalviar temps de processament al microcontrolador.

Per tal d'activar la localització per GPS posarem crearem un seguit de funcions que activarem únicament quan es doni una situació de caiguda. Les dades del GPS després de la seva activació triguen una estona a estar disponibles, per tant s'ha de crear un algorisme bloquejant que es quedi en bucle fent les consultes al GPS per tal de poder extreure una localització adequada. Això ho farem a partir del següent programa.

- **Programa 6 Alarma de Caiguda i Localització GPS.**

```

#include "GY_85.h"
#include <Wire.h>
#include <SD.h>
#include <string.h>
#include <TinyGPS++.h>
#include <SoftwareSerial.h>
#define Rad 180/PI

GY_85 GY85;      //create the object

SoftwareSerial GPS(8,9);//RX,TX Comunicació amb el GPS
TinyGPSPlus gps;//OBJECTE GPS

```



```
int RXLED = 17;
File Fitxer_dades;

String crea_fitxer(){
    bool fitxer_existent=true;
    int numero_int=0;
    String numero_str=(String) numero_int;
    String Nom_fitxer="Fitxer"+numero_str+".txt";

    while(fitxer_existent){
        if(SD.exists(Nom_fitxer)){
            numero_int++;
            numero_str=(String) numero_int;
            Nom_fitxer="Fitxer"+numero_str+".txt";
        }
        else{
            Fitxer_dades=SD.open(Nom_fitxer, FILE_WRITE);//Creem el fitxer
            numero X
            Fitxer_dades.close();
            fitxer_existent=false;
        }
    }
    return Nom_fitxer;
}
bool displayInfo()
{
    static int contador=0;

    if (gps.location.isValid())//LOCALITZACIÓ
    {
        Serial.print(F("Satelit: "));
        Serial.print(gps.satellites.value());
    }
}
```

```
Serial.print(F(", Localitzacio: "));  
Serial.print(gps.location.lat(),10);  
Serial.print(" , ");  
Serial.println(gps.location.lng(),10);  
contador++;  
}  
else  
{  
    Serial1.print(F("INVALID"));  
}  
Serial1.print(F("  Data/hora: "));  
if (gps.date.isValid())//DATA  
{  
    Serial1.print(gps.date.month());  
    Serial1.print(F("/"));  
    Serial1.print(gps.date.day());  
    Serial1.print(F("/"));  
    Serial1.print(gps.date.year());  
}  
else  
{  
    Serial1.print(F("INVALID"));  
}  
Serial1.print(" ");  
if (gps.time.isValid())//HORA  
{  
    if (gps.time.hour() < 10) Serial1.print(F("0"));  
    Serial1.print(gps.time.hour());
```

```
    Serial1.print(F(":"));

    if (gps.time.minute() < 10) Serial1.print(F("0"));

    Serial1.print(gps.time.minute());

    Serial1.print(F(":"));

    if (gps.time.second() < 10) Serial1.print(F("0"));

    Serial1.println(gps.time.second());

}

else

{

    Serial1.println(F("INVALID"));

}

if(contador>9){

    Serial.println(contador);

    contador=0;

    return false;

}

return true;

}

void alarma_GPS(){

    bool semaf=true;

    while(semaf){

        while(GPS.available()){

            if (gps.encode(GPS.read()))

                semaf=displayInfo();

        }

    }

}

void caiguda(){
```

```
Serial1.println("S'ha petit una caiguda!!");
alarma_GPS();
}

void setup()
{
    pinMode(RXLED, OUTPUT);
    //inicialitza la comunicació serial
    Serial.begin(9600);
    //inicialitza la comunicació serial1
    Serial1.begin(9600);
    //inicialitza la comunicació serial software amb el GPS
    GPS.begin(9600);
    // conf GY-85
    Wire.begin();
    delay(4);
    GY85.init();
    delay(4);
    if(!SD.begin(A0)){
        Serial.println("No s'ha pogut inicialitzar");
        return;
    }
    Serial.println("inici ok");
    tiempo_prev=millis();
}

void loop()
{
    int temps=5;
    float ax, ay, az;
    float gx, gy, gz;
    float ax_0, ay_0, az_0=0;
    bool primera_volta=false;
    float accel_ang_y_0=0;
    float ang_y, ang_y_prev;
    String Nom_fitxer=crea_fitxer();
    while(true){
```

```

//posada a zero a l'arrencada
if(primer_volta==false){
    ax_0=GY85.accelerometer_x(GY85.readFromAccelerometer())*0.082;
    ay_0=GY85.accelerometer_y(GY85.readFromAccelerometer())*0.082;
    az_0=GY85.accelerometer_z(GY85.readFromAccelerometer())*0.082;

    accel_ang_y_0=atan(ay_0/az_0)*(Rad);
    primera_volta=true;
}

//lectura dels sensors
ax = GY85.accelerometer_x( GY85.readFromAccelerometer() )*0.082;
ay = GY85.accelerometer_y( GY85.readFromAccelerometer() )*0.082;
az = GY85.accelerometer_z( GY85.readFromAccelerometer() )*0.082;
gx = GY85.gyro_x( GY85.readGyro() );
gy = GY85.gyro_y( GY85.readGyro() );
gz = GY85.gyro_z( GY85.readGyro() );
dt=(millis()-tiempo_prev)/1000.0;
tiempo_prev=millis();
//Calcul de l'angle amb l'acceleròmetre
float accel_ang_y=atan(ay/az)*(Rad)-accel_ang_y_0;
//calcul angle amb giroscopi
float giro_y=ang_y_prev+(gy)*dt;
//filtre complementari
ang_y = 0.7*(giro_y) + 0.3*accel_ang_y;
Serial.println(ang_y);//Transmissió inalàmbrica
Serial1.println(ang_y);//Transmissió inalàmbrica
// escriptura de dades a la SD
Fitxer_dades=SD.open(Nom_fitxer, FILE_WRITE);//obrim el fitxer en mode
lectura
if(Fitxer_dades){
    Fitxer_dades.print(tiempo_prev);
    Fitxer_dades.print(F("\t"));
    Fitxer_dades.print(ay);
    Fitxer_dades.print(F("\t"));
    Fitxer_dades.print(az);
}

```

```
Fitxer_dades.print(F("\t"));
Fitxer_dades.print(gy);
Fitxer_dades.print(F("\t"));
Fitxer_dades.print(giro_y);
Fitxer_dades.print(F("\t"));
Fitxer_dades.println(ang_y);
Fitxer_dades.close();//tanquem l'arxiu
}
else{
    Serial.println(F("error al obrir l'arxiu"));
}
if (ang_y> 65 || ang_y<-65){
    caiguda();
    Serial.println("Alarma");
}
ang_y_prev=ang_y;

}
}
```

2.3. Proves i treball de camp.

En aquest apartat es pot veure els diferents testos que s'han efectuat amb el prototip. Així, podem veure els diferents assajos efectuats per tal d'establir les característiques de l'aparell fabricat, i ajustar-lo dins del possible per tal de poder efectuar la tasca que li volem encomanar, en aquest cas detectar les inclinacions en una motocicleta, i donar una alarma si aquesta ha patit una caiguda.

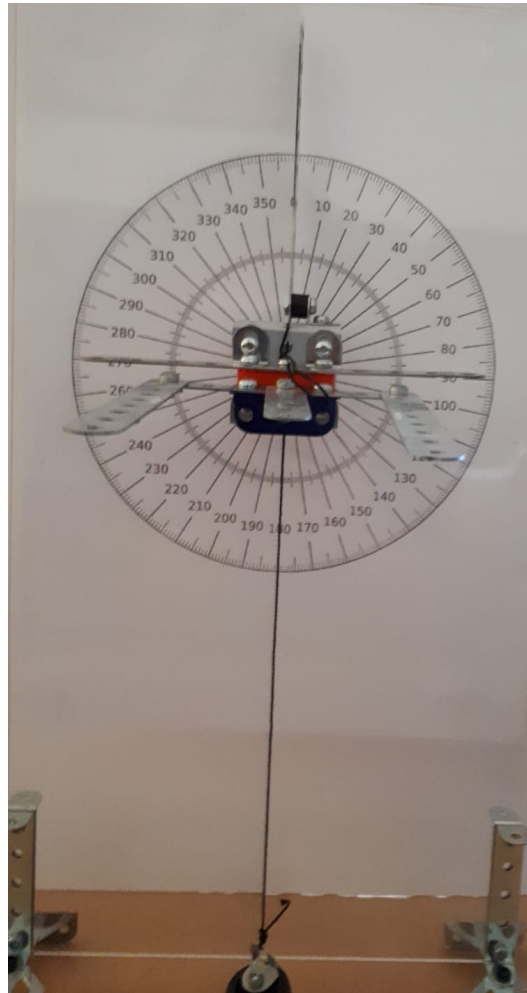


Figura 38: Instrument de mesura d'angles.

2.3.1 Prova 1 Detecció de la inclinació absoluta i relativa.

Per poder detectar la inclinació absoluta de l'aparell respecte la gravetat terrestre, s'utilitza el Programa 3: Detecció dels angles i implementació del filtre complementari, sense fer la resta de la inclinació inicial.

```
float accel_ang_y=atan(ay/az)*(Rad); //-accel_ang_y_0;
```

Per tal de poder determinar quin angle té el prototip construït en cada moment, es fa la construcció d'un instrument de mesura per tal de poder determinar quin angle té el prototip en cada moment.

En la figura 37, podem apreciar l'instrument de mesura d'angles fabricat. Aquest està fet amb un panell de metacrilat, un full amb un transportador d'angles de 360° imprès, una estructura feta de plaques d'acer i plàstic degudament collades. En l'estructura podem distingir una placa totalment horitzontal per tal de marcar els 90° i els 270°. Perpendicularment a aquesta placa horitzontal podem apreciar una segona placa totalment vertical la qual en la figura marca els 0°. quan la nostre estructura comenci a moure's, la placa vertical ens marcarà l'angle d'inclinació en cada moment. En la part inferior podem apreciar un pendol, aquest marca la direcció de la gravetat terrestre. Així quan aquest es quedi alineat amb els 180° del transportador d'angles vol dir que el transportador el tenim situat en la posició correcte. En aquest punt és moment de fixar-lo al metacrilat per tal d'evitar moviments en aquest. En la part davantera podem veure 3 plaques en mirant-nos, aquestes plaques serviran per poder situar el nostre prototip.

En aquest cas el sensor encarregat de detectar la força de la gravetat és l'acceleròmetre, i per tant anem a fer una prova només utilitzant aquest sensor. Per tal de que el giroscopi no interfereixi en la nostre mesura retocarem els valors del filtre complementari.

```
ang_y = 0*(giro_y) + 1*accel_ang_y;
```

Un cop carregat el programa, col·loquem el dispositiu sobre el suport del nostre mesurador d'angles.

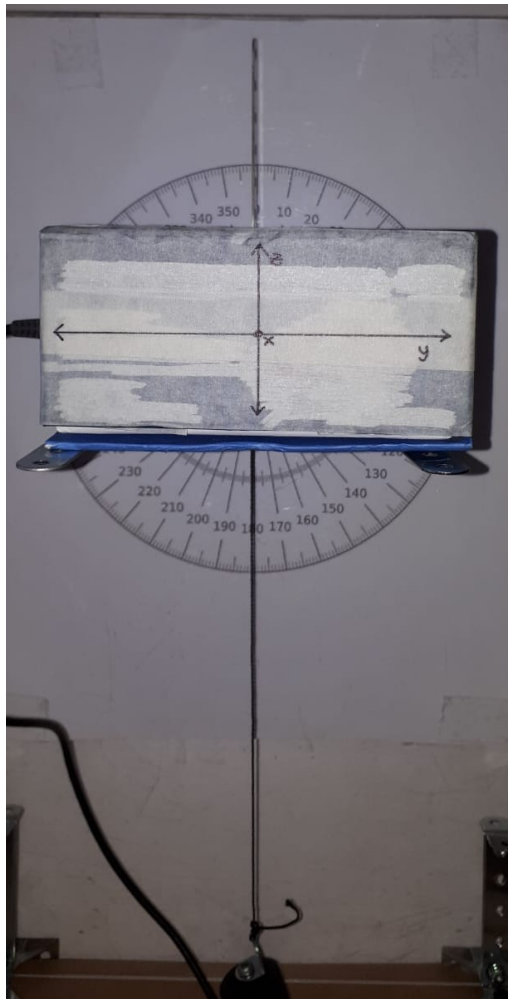


Figura 39: Mesura absoluta dels 0° amb l'acceleròmetre.

| COM11 | |
|--------------|---------|
| | |
| 13:42:18.481 | -> 3.88 |
| 13:42:18.515 | -> 3.39 |
| 13:42:18.582 | -> 3.88 |
| 13:42:18.650 | -> 3.88 |
| 13:42:18.684 | -> 3.42 |
| 13:42:18.717 | -> 3.85 |
| 13:42:18.786 | -> 3.42 |
| 13:42:18.819 | -> 3.39 |
| 13:42:18.888 | -> 3.42 |
| 13:42:18.921 | -> 3.37 |
| 13:42:18.989 | -> 3.88 |
| 13:42:19.023 | -> 3.37 |
| 13:42:19.090 | -> 3.39 |
| 13:42:19.124 | -> 3.88 |
| 13:42:19.225 | -> 2.91 |
| 13:42:19.462 | -> 3.39 |
| 13:42:19.530 | -> 3.88 |
| 13:42:19.564 | -> 3.39 |

Figura 40: Valors del prototip en graus.

En la figura 38, podem veure el nostre prototip sobre l'aparell de mesura. Es pot apreciar en la imatge com els 0° i els 180° es troben perfectament alineats amb l'eix Z, i com l'eix Y és perpendicular a aquests. Per tant anem a mesurar l'angle que ens dona el prototip amb una mesura absoluta a partir de la força de la gravetat.

En la figura 39, es pot veure els valors retornats per el prototip a partir del port Serial. Com es pot veure, ens retorna un valor mitjà de $3,55^\circ$. Per tant podem concloure que l'acceleròmetre dins la caixa no es troba en una posició totalment horitzontal i que per tant té un error o deriva aproximada de $3,55^\circ$.

Cal mencionar que les mesures de l'acceleròmetre les estem efectuant amb una sensibilitat de 4G.

Per solucionar aquest problema hem fet un algoritme el qual mesura la inclinació relativa de l'aparell, i no l'absoluta, ja que en l'absoluta tenim una deriva de $3,55^\circ$ quan realment ens trobem a 0° .

Aquest algoritme és el que es pot apreciar en el programa Programa 3: Detecció dels angles i implementació del filtre complementari. Aquest programa a l'inici de l'algoritme llegeix la posició de l'aparell respecte la gravetat terrestre. Aquest angle serà utilitzat com a "angle 0" respecte el terra i serà restat en la detecció de l'angle per l'acceleròmetre. Per tant si l'acceleròmetre al iniciar el programa detecta que està a 5° respecte el terra, posteriorment aquests 5° seran restats per tal d'aconseguir la posició relativa del nostre aparell respecte el terra. Imaginem que no movem l'aparell, així en la detecció de l'angle en l'acceleròmetre tornarà a aparèixer 5° per tant l'angle total detectat per l'acceleròmetre en el prototip serà $5^\circ - 5^\circ = 0^\circ$.

```
float accel_ang_y=atan(ay/az)*(Rad)-accel_ang_y_0;
```

Sense moure el prototip en el suport carreguem el programa 3 complet sense modificacions, per tant amb la resta en el càlcul de l'angle en l'acceleròmetre per extreure la inclinació relativa a partir d'aquest sensor.

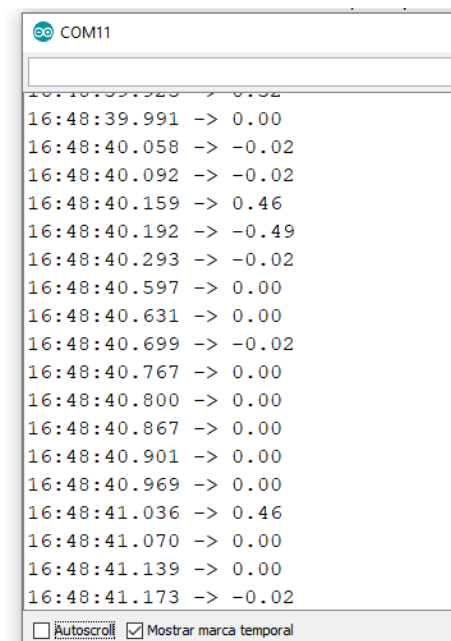


Figura 41: Angle relatiu a partir de l'acceleròmetre.

En la figura 40, podem veure els graus de inclinació relativa de l'aparell. Com que l'aparell de mesura està situat a 0° , esperem un valor de 0° , tot i això, podem veure que en les lectures hi ha alguns errors produïts segurament per petites vibracions o per la mateixa lectura del sensor. Tot i això podem extreure una mitjana de $0,0183^\circ$ amb els valors de la figura 40. Per tant la desviació es pot considerar menyspreable en el cas de l'acceleròmetre.

Efectuar aquesta prova per el giroscopi és innecessari, ja que aquest només té sentit quan el prototip ja ha tingut algun moviment.

2.3.2 Prova 2 Detecció de la inclinació en estàtic, detecció de 10°.

2.3.2.1 Acceleròmetre.

```
ang_y = 0*(giro_y) + 1*accel_ang_y;
```

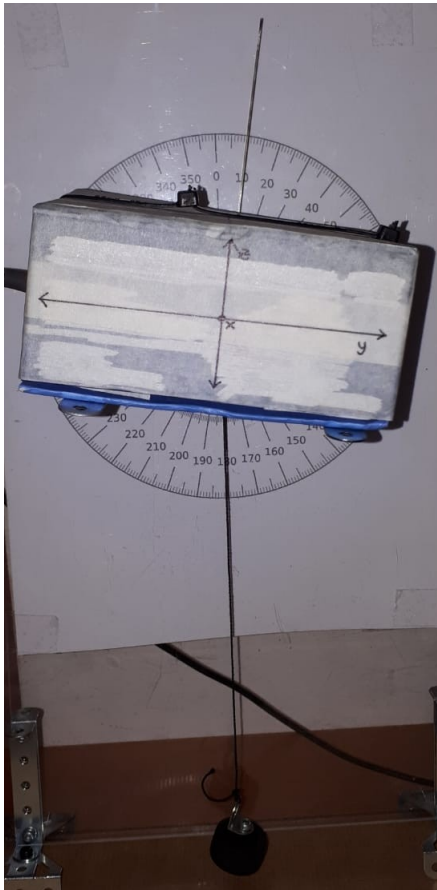


Figura 42: Mesura de 10° amb l'acceleròmetre.

| COM11 | |
|--------------|----------|
| 17:20:43.933 | -> 10.72 |
| 17:20:44.001 | -> 11.19 |
| 17:20:44.034 | -> 10.72 |
| 17:20:44.134 | -> 11.19 |
| 17:20:44.169 | -> 10.60 |
| 17:20:44.237 | -> 11.19 |
| 17:20:44.271 | -> 10.72 |
| 17:20:44.339 | -> 10.60 |
| 17:20:44.407 | -> 10.72 |
| 17:20:44.474 | -> 11.06 |
| 17:20:44.542 | -> 10.60 |
| 17:20:44.576 | -> 10.60 |
| 17:20:44.915 | -> 10.72 |
| 17:20:44.948 | -> 10.72 |
| 17:20:45.016 | -> 11.19 |
| 17:20:45.050 | -> 10.72 |
| 17:20:45.117 | -> 10.72 |
| 17:20:45.185 | -> 10.72 |

Figura 43: Mesura de 10° amb l'acceleròmetre.

En la figura 41, podem veure el prototip inclinat 10° a partir de l'aparell de mesura estàtica. En aquesta mesura l'únic sensor que s'està tenint en compte és l'acceleròmetre. Per efectuar aquesta mesura, l'acceleròmetre ha estat calibrat a 4G i el filtre complementari només té en compte la mesura d'aquest, desestima completament la mesura feta per el giroscopi, per tant en la figura 42 podem veure els graus de inclinació que ens retorna el prototip. La mitjana dels valors retornats és de 10,72°, i la desviació en la lectura no és molt elevada de 0.228° , també és cert que és relativament complicat saber si la posició del mesurador és de 10° justos o ja anem una mica cap a 11°.

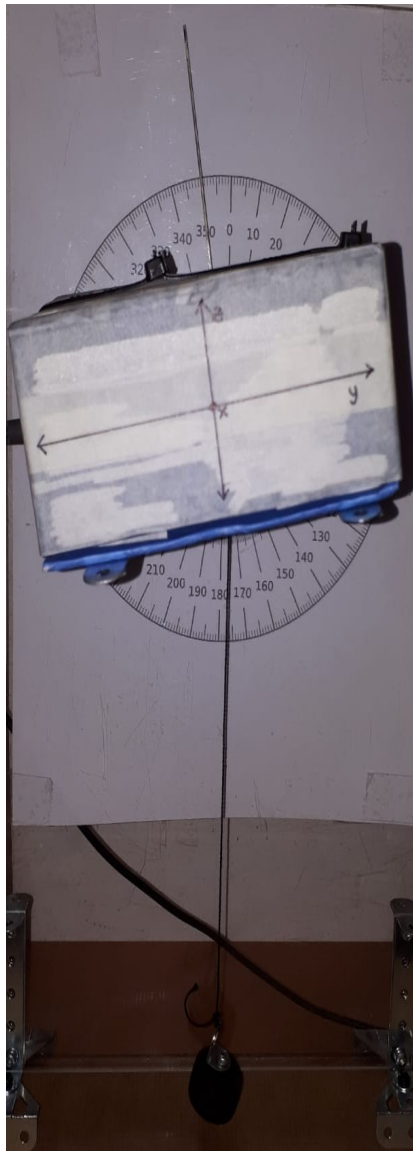


Figura 44: Mesura de -10° amb l'acceleròmetre.

| COM11 | | |
|--------------|----|--------|
| 17:37:24.417 | -> | -10.76 |
| 17:37:24.486 | -> | -11.19 |
| 17:37:24.520 | -> | -10.76 |
| 17:37:24.588 | -> | -11.67 |
| 17:37:24.926 | -> | -10.07 |
| 17:37:24.994 | -> | -10.27 |
| 17:37:25.028 | -> | -10.70 |
| 17:37:25.094 | -> | -11.24 |
| 17:37:25.162 | -> | -10.76 |
| 17:37:25.195 | -> | -10.70 |
| 17:37:25.263 | -> | -10.70 |
| 17:37:25.330 | -> | -10.76 |
| 17:37:25.400 | -> | -10.70 |
| 17:37:25.434 | -> | -11.19 |
| 17:37:25.503 | -> | -10.22 |
| 17:37:25.538 | -> | -11.19 |
| 17:37:25.678 | -> | -10.17 |
| 17:37:25.949 | -> | -11.73 |

Figura 45: Mesura de -10° amb l'acceleròmetre.

Tal i com es pot veure en les figures 43 i 44, repetim la mateixa mesura per valors negatius. Podem comprovar a partir dels valors de la figura 44, que l'error en aquest cas és pràcticament el mateix que per els valors positius. En aquest cas el valor mitjà ens dona $10,76^\circ$ i la desviació de les lectures és de $0,47^\circ$. Per tant en aquest cas seguim tenint un error petit.

Podem considerar que la mesura amb l'acceleròmetre és estable en temps llargs.

Si repetim la operació 20 vegades anant de 10° a -10° , els valors detectats per l'acceleròmetre no varien en absolut, sempre són els mateixos o pràcticament iguals.

2.3.2.2 Giroscopi.

En aquest cas repetim la mateixa prova que en les figures 41 i 43, però en aquest cas no tenim en compte l'acceleròmetre, això ho fem a partir de retocar el filtre complementari, per tant donarem tot el pes de la detecció al giroscopi i menystindrem la detecció de l'acceleròmetre.

```
ang_y = 1*(giro_y) + 0*accel_ang_y;
```

Posem el nostre sistema de detecció a 0° , un cop carregat el programa, quan s'obre el port serial es pot observar que l'angle detectat és 0° en totes les lectures, ja que el sistema encara no ha petit moviment.

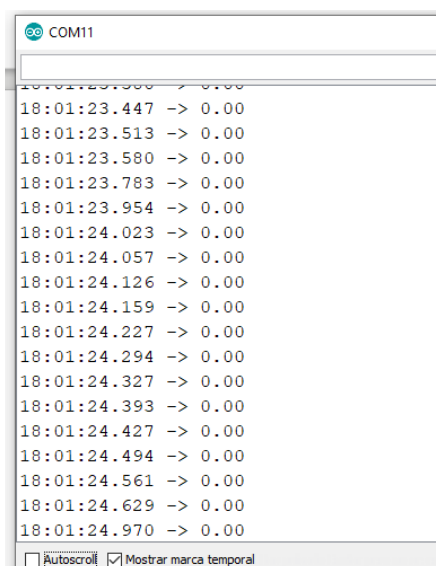


Figura 46: Mesura de 0° amb el giroscopi.

Tal i com fem en l'apartat anterior de l'acceleròmetre, posem la plataforma amb el sensor a 10° tal i com es mostra en la figura 41. En aquest cas els valors retornats per el giroscopi són els següents:

| Timestamp | Value |
|--------------|---------|
| 18:53:03.366 | -> 7.77 |
| 18:53:03.535 | -> 7.77 |
| 18:53:03.569 | -> 7.77 |
| 18:53:03.603 | -> 7.77 |
| 18:53:03.670 | -> 7.77 |
| 18:53:03.703 | -> 7.77 |
| 18:53:03.737 | -> 7.77 |
| 18:53:03.772 | -> 7.77 |
| 18:53:03.806 | -> 7.77 |
| 18:53:03.841 | -> 7.77 |
| 18:53:03.875 | -> 7.77 |
| 18:53:03.943 | -> 7.77 |
| 18:53:03.980 | -> 7.77 |
| 18:53:04.011 | -> 7.77 |
| 18:53:04.046 | -> 7.77 |
| 18:53:04.216 | -> 7.77 |
| 18:53:04.553 | -> 7.77 |
| 18:53:04.586 | -> 7.77 |

Figura 47: Mesura de 10° amb el giroscopi intent 1.

| Timestamp | Value |
|--------------|----------|
| 19:04:32.730 | -> 11.15 |
| 19:04:32.763 | -> 11.15 |
| 19:04:32.831 | -> 11.15 |
| 19:04:32.866 | -> 11.15 |
| 19:04:32.901 | -> 11.15 |
| 19:04:32.936 | -> 11.15 |
| 19:04:32.971 | -> 11.15 |
| 19:04:33.005 | -> 11.15 |
| 19:04:33.072 | -> 11.15 |
| 19:04:33.279 | -> 11.15 |
| 19:04:33.553 | -> 11.15 |
| 19:04:33.586 | -> 11.15 |
| 19:04:33.620 | -> 11.15 |
| 19:04:33.654 | -> 11.15 |
| 19:04:33.720 | -> 11.15 |
| 19:04:33.754 | -> 11.15 |
| 19:04:33.787 | -> 11.15 |

Figura 48: Mesura de 10° amb el giroscopi intent 2.

| Timestamp | Value |
|--------------|----------|
| 19:06:31.904 | -> 13.62 |
| 19:06:32.305 | -> 13.62 |
| 19:06:32.305 | -> 13.62 |
| 19:06:32.305 | -> 13.62 |
| 19:06:32.382 | -> 13.62 |
| 19:06:32.517 | -> 13.62 |
| 19:06:32.551 | -> 13.62 |
| 19:06:32.586 | -> 13.62 |
| 19:06:32.654 | -> 13.62 |
| 19:06:32.688 | -> 13.62 |
| 19:06:32.722 | -> 13.62 |
| 19:06:32.755 | -> 13.62 |
| 19:06:32.788 | -> 13.62 |
| 19:06:32.855 | -> 13.62 |
| 19:06:32.888 | -> 13.62 |
| 19:06:32.922 | -> 13.62 |
| 19:06:32.955 | -> 13.62 |

Figura 49: Mesura de 10° amb el giroscopi intent 3.

El que podem apreciar amb el giroscopi, és que l'angle que aquest detecta depèn de la velocitat amb la qual girem el nostre sistema, en la figura 46 i primer intent hem girat el sistema de 0 fins a 10° amb una velocitat baixa, en els següents intents hem girat el sistema amb una velocitat cada cop més alta, com podem apreciar en les figures 47 i 48, cada cop l'error és diferent. El problema s'amplifica quan volem retornar el sensor a la seva posició de repòs a 0°.

| Timestamp | Value |
|--------------|---------|
| 19:22:29.789 | -> 3.42 |
| 19:22:29.824 | -> 3.42 |
| 19:22:29.858 | -> 3.42 |
| 19:22:29.892 | -> 3.42 |
| 19:22:29.959 | -> 3.42 |
| 19:22:29.993 | -> 3.42 |
| 19:22:30.197 | -> 3.42 |
| 19:22:30.500 | -> 3.42 |
| 19:22:30.535 | -> 3.42 |
| 19:22:30.568 | -> 3.42 |
| 19:22:30.602 | -> 3.42 |
| 19:22:30.670 | -> 3.42 |
| 19:22:30.703 | -> 3.42 |
| 19:22:30.738 | -> 3.42 |
| 19:22:30.772 | -> 3.42 |
| 19:22:30.840 | -> 3.42 |
| 19:22:30.873 | -> 3.42 |

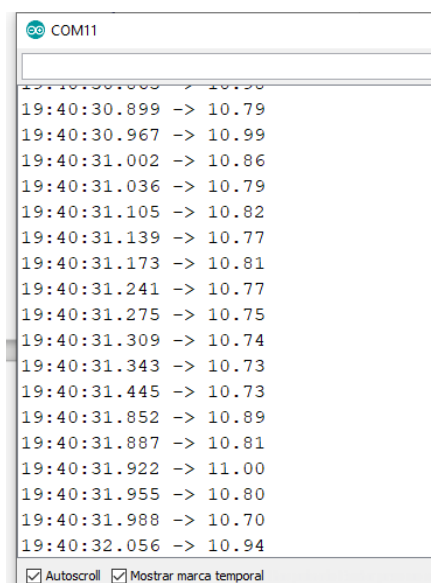
Figura 50: Retorn de 10° a 0° amb el giroscopi.

Podem veure en la figura 49 que en fer el retorn de 10° a 0° el valor proporcionat per el giroscopi no és l'esperat, i aquest té un error important. D'aquest error se'n diu DRIFT. I és el motiu per el qual el giroscopi per ell mateix no ens serveix per realitzar mesures en un temps prolongat, ja que aquests errors s'anirien sumant i acabarien afectant molt perjudicialment la mesura.

D'aquesta forma podem concloure que el giroscopi per ell mateix no ens serveix per realitzar mesures, degut al drift, però com podem observar les lectures d'aquest són poc variants, estables ja que la desviació de les lectures és igual a 0°. Així que aquest ens pot servir per estabilitzar les lectures del nostre equip.

2.3.2.3 Filtre complementari Acceleròmetre 50% Giroscopi 50%.

Tal i com hem fet en els apartats anteriors, fem la prova de la inclinació dels 10° . Però ara ajuntant els dos sensors amb el filtre complementari, donarem un pes del 50% en el cas del Giroscopi i un 50% del pes en el cas de l'acceleròmetre per veure com reaccionen junts.



Com podem veure en la figura 50, la mesura està una mica desplaçada, però la dispersió entre els punts és menor. En la mesura tenim una valor mitjà de $10,79^\circ$, i una desviació en la lectura dels valors de $0,087^\circ$, molt menor que en el cas de l'acceleròmetre on aquesta havia arribat fins a $0,47^\circ$. Per tant podem concloure que la detecció d'angles a partir del filtre complementari és molt millor que utilitzar qualsevol dels altres dos sensors sols.

Figura 51: Mesura de 10° acceleròmetre i giroscopi.

A partir del programa Octave i la targeta de memòria hem fet uns petits scripts els quals ens ajuden a l'hora d'interpretar els valors obtinguts a partir de gràfiques. D'aquesta forma a part de veure l'evolució dels valors en el temps podem comparar els valors dels sensors amb els angles obtinguts. A la vegada també podem veure els temps en els quals estem treballant, temps d'execució, retards, etc...

El programa d'interpretació dels valors obtinguts en Octave és el següent:

- **Programa Octave Tractament de Dades.**

```
file=load('Prova_50.TXT');
temps=file(:,1)/1000;# en segons
ay=file(:,2);
az=file(:,3);
gy=file(:,4);
giro_y=file(:,5);
ang_y=file(:,6);
figure(1);
plot(temps, ay,"r",temps, az,"g",temps, ang_y,"b");
figure(2);
plot(temps, gy,"r",temps, giro_y,"g",temps, ang_y,"b");
```

En el programa anterior podem apreciar els valors guardats en la targeta de memòria. Aquests valors estan guardats en forma de columnes les quals podem extreure fàcilment i convertir en matrius de valors. El programa carrega tot el fitxer llegit en la variable file, a partir d'aquest punt extraïem cada una de les columnes amb la forma variable=file(:,columna a extreure), d'aquesta forma posem cada una de les columnes del fitxer a una variable, aquestes són:

- ay→ Valor de l'acceleròmetre en Y.
- az→ Valor de l'acceleròmetre en Z.
- gy→ valor del giroscopi en Y.
- giro_y→ angle calculat per el giroscopi.
- ang_y→ angle total aplicant el filtre complementari.

Posteriorment podem veure 2 gràfiques.

En la primera gràfica, tractem els valors obtinguts amb l'acceleròmetre, acceleració en Y de color vermell, acceleració en Z de color verd, i angle total obtingut de color blau.

En la segona gràfica, tractem els valors obtinguts per el giroscopi en l'eix Y de color vermell, l'angle calculat per el giroscopi de clor verd , i l'angle total obtingut de color blau.

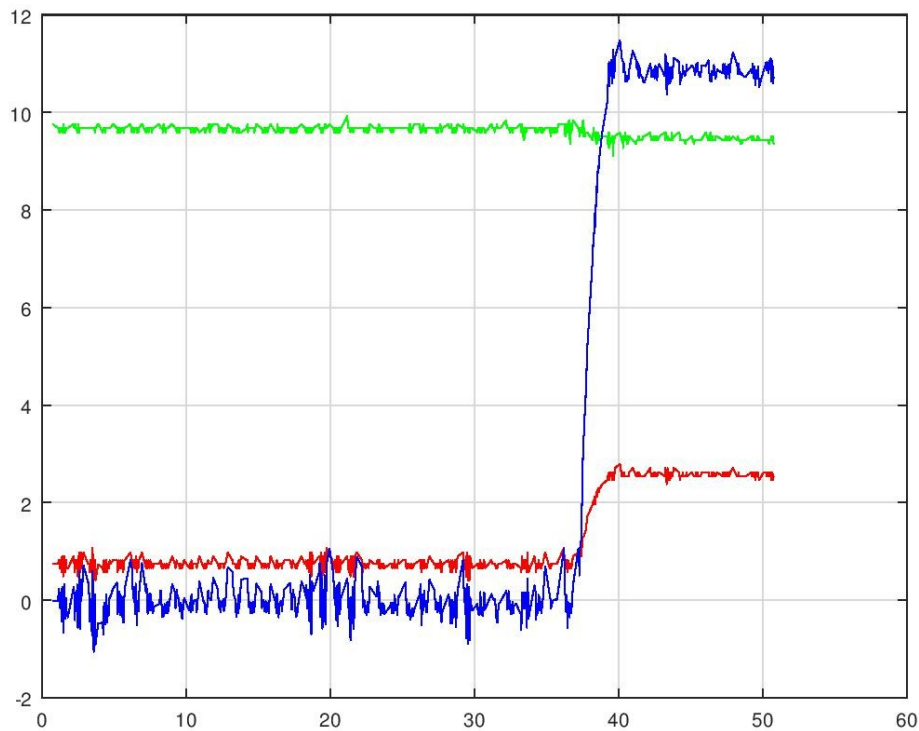


Figura 52: Gràfica 1, inclinació 10° acceleròmetre.

En la figura 51, podem veure les acceleracions captades per l'acceleròmetre en l'eix Y (verd) i el l'eix Z (vermell). Com podem apreciar, l'eix Y, té una acceleració aproximada de 9.8 m/s^2 , i l'eix Z té una acceleració de 0.7 m/s^2 quan el prototip està a 0° . Podem veure que aproximadament als 37 segons pateix un canvi, aquest canvi va relacionat amb el moviment del prototip des de 0° fins a 10° . Com es pot veure, el canvi succeeix a una velocitat determinada, i es pot veure clarament la progressió que pateixen els valors. En l'eix Y, pràcticament no observem variació, però en l'eix Z aquesta sí que és molt notòria. De color blau podem veure els valors de l'angle total calculat. Es pot veure que aquest té un soroll en 0° , el qual ve determinat per el soroll que produeix l'acceleròmetre, ja que tant en l'eix Y com en l'eix Z podem veure que les lectures de les acceleracions són fluctuants. quan a partir d'aquests dos es calcula l'angle, si aquest calcul agafa 2 màxims de soroll aquest es converteix en un valor considerable. quan en el segon 37 fem el canvi de 0° a 10° , podem veure que la progressió de valors és nítida, i que un cop estabilitzada la lectura de l'acceleròmetre,

l'angle també s'estabilitza. En la gràfica no es pot apreciar visualment cap retard entre la variació en les acceleracions i l'angle calculat per el filtre.

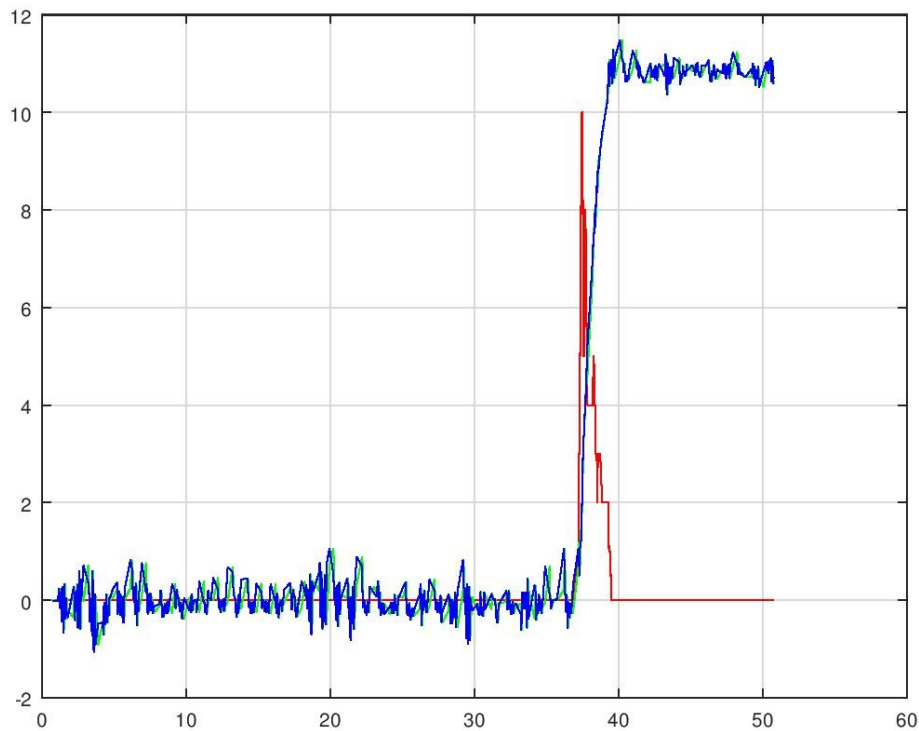


Figura 53: Gràfica 2, inclinació 10° giroscopi.

En la gràfica 52, podem veure les rotacions captades per el giroscopi en color vermell. Podem apreciar que el valor del giroscopi és 0°/s quan el dispositiu té absència de moviment. quan el dispositiu té moviment podem veure com aquest modifica el seu valor. Cal recordar que el que podem veure en vermell són graus/segon, per tant no és la inclinació total. En verd podem veure el càlcul de l'angle segons el giroscopi, i en blau podem veure el calcul de l'angle segons el filtre complementari. Si ens fixem en el programa 3 podem apreciar la línia 1 el calcul del valor del giroscopi, aquest aprofita l'angle guardat en la volta anterior el qual podem veure en la línia 2. Aquest angle de la línia 2, és l'angle calculat per el filtre complementari, per tant és normal que la línia verda segueixi tant de prop la blava, ja que en el calcul de l'angle del giroscopi s'aprofita el valor anterior calculat per el filtre.

```
1. giro_y=ang_y_prev+(gy)*dt;
```

```
2. ang_y_prev=ang_y;
```

En aquest cas podem concloure que el filtre aplica molt de soroll a la mesura ja que la ponderació dels sensors pot ser no és l'adequada.

2.3.3 Prova 3 Detecció de la inclinació en estàtic, detecció de 90° i retorn a 0°.

Per efectuar aquesta prova repetirem el mateix que en l'apartat 2.3.2.3 Filtre complementari Acceleròmetre 50% Giroscopi 50%, però en aquest cas, anirem incrementant la inclinació de 10° en 10° fins a arribar a 90°. Per fer-ho desactivarem la detecció d'alarma per tal de que es puguin emmagatzemar els valors adequadament. Un cop detectats els 90° retornarem a 0° i efectuarem la mateixa prova per els valors negatius. Ens fixarem en els graus que marca el port serial en cada cas per cada progressió, i posteriorment tractarem les dades amb el programa Octave vist en l'apartat anterior.

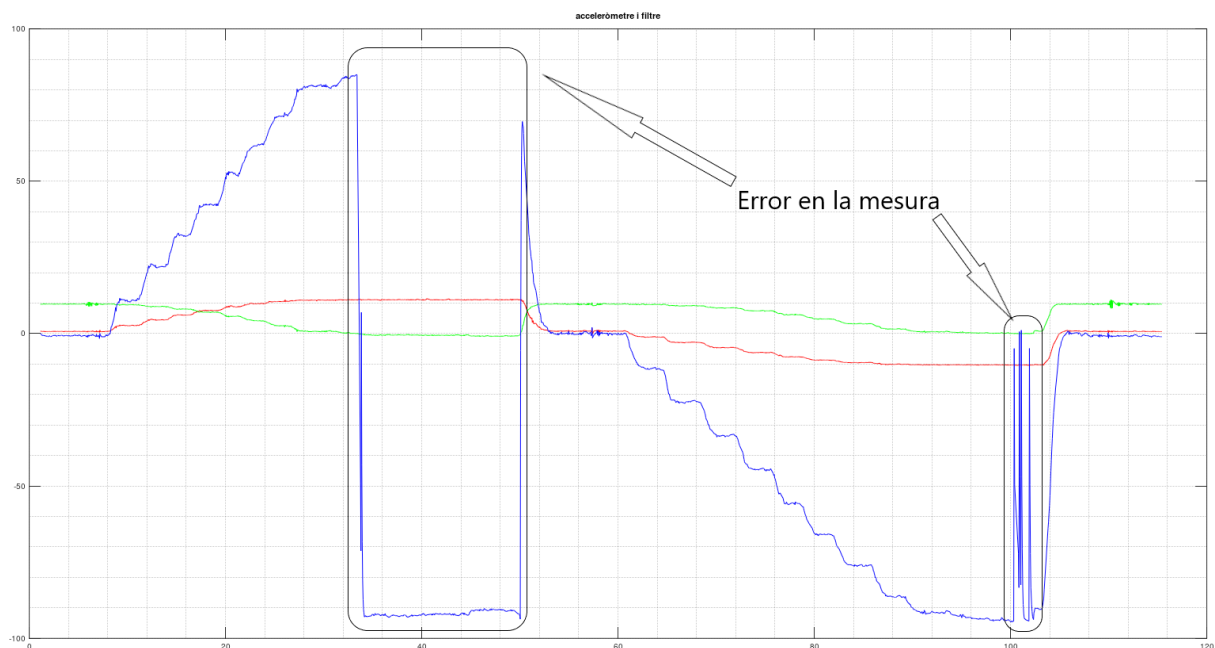


Figura 54: Gràfica 1 acceleròmetre i filtre +90° -> 0° -> -90°-> 0°

En la figura 53 podem veure la prova dels 90° efectuada. En aquesta s'ha anat incrementant l'angle de 10° en 10° fins a assolir els 90°. Cal dir que s'ha pres com a mesura dels angles el mesurador d'angles fabricat, per tant en la gràfica podem veure com evolucionen els valors detectats per el prototip. Podem veure que la zona on l'error de mesura és més gran és en els angles mitjans concretament el més gran el trobem a 50° on el valor mesurat arriba fins a 53°. Per tant podem dir que el nostre aparell té un error màxim de mesura de 3°. El més preocupant és que en aproximar-nos a 90° succeeix un error en la mesura el qual en comptes de donar 90° dona -90°. Aquest fet es repeteix quan assolim els -90° quan estem efectuant la prova negativa. El que detecta el sistema és que comença a estar al revés i per tant la

formula de l'arc tangent deixa de ser eficaç. Tot i això una caiguda en una motocicleta es dona a partir de menys graus, per tant aquest fet no es significatiu per l'aplicació d'aquest projecte, ja que en aquest cas detectarem una caiguda sempre i quan la motocicleta estigui per sobre de 65° o per sota de -65° .

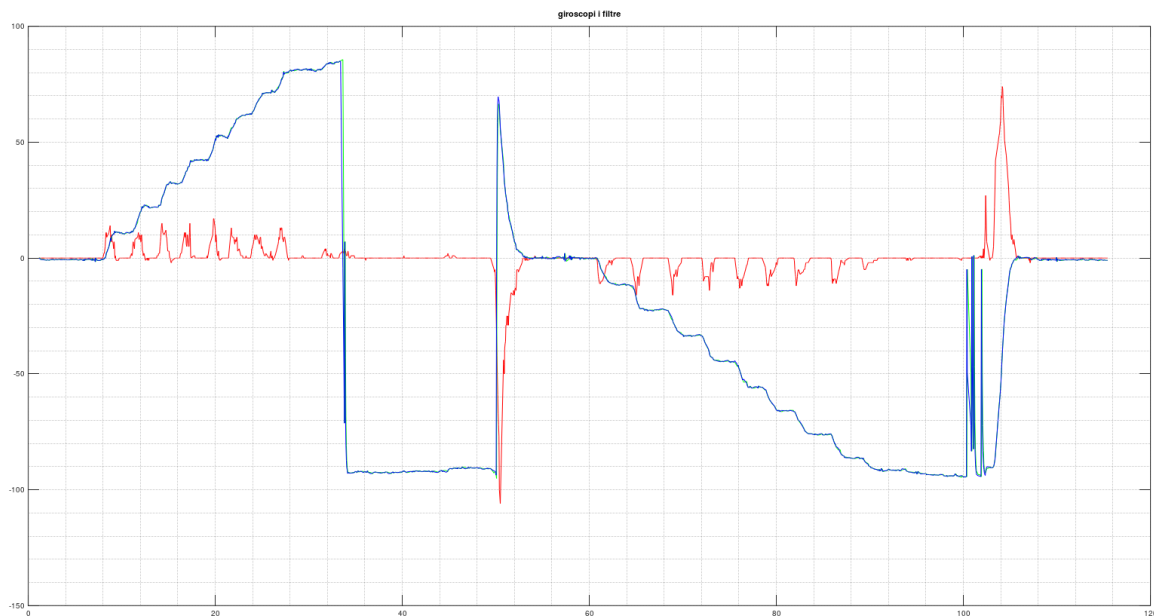


Figura 55: Gràfica 2 giroscopi i filtre $+90^\circ \rightarrow 0^\circ \rightarrow -90^\circ \rightarrow 0^\circ$

En aquest cas en la gràfica 54 podem veure la gràfica dels angles amb les rotacions pertinents en cada moment de rotació. Podem veure que en aquest cas no s'ha aplicat una rotació gaire ràpida al sistema. També podem veure que la gràfica dels valors del giroscopi i la del filtre són pràcticament invisibles ja que l'angle calculat per el filtre realimenta al calcul de l'acceleròmetre.

2.3.4 Prova 4 Detecció de la inclinació en estàtic, filtre 98% giroscopi 2% acceleròmetre.

En l'apartat anterior hem vist que el nostre sistema tenia soroll per culpa de l'acceleròmetre. Per que aquest soroll sigui inferior s'utilitza l'estratègia de ponderar el càlcul efectuat per el giroscopi amb més pes que el càlcul efectuat per el l'acceleròmetre. Es decideix provar amb els valors de 98% per el giroscopi i un 2% per l'acceleròmetre per veure les reaccions d'aquests. En aquest cas aquesta ponderació del filtre complementari actuarà com a filtre passa alt per el giroscopi i de filtre passa baixos per l'acceleròmetre. Així aquest ultim només l'utilitzarem per estabilitzar la mesura un cop hagui passat un temps relativament llarg i el que tindrà pes a curt termini són els valors calculats per el giroscopi.

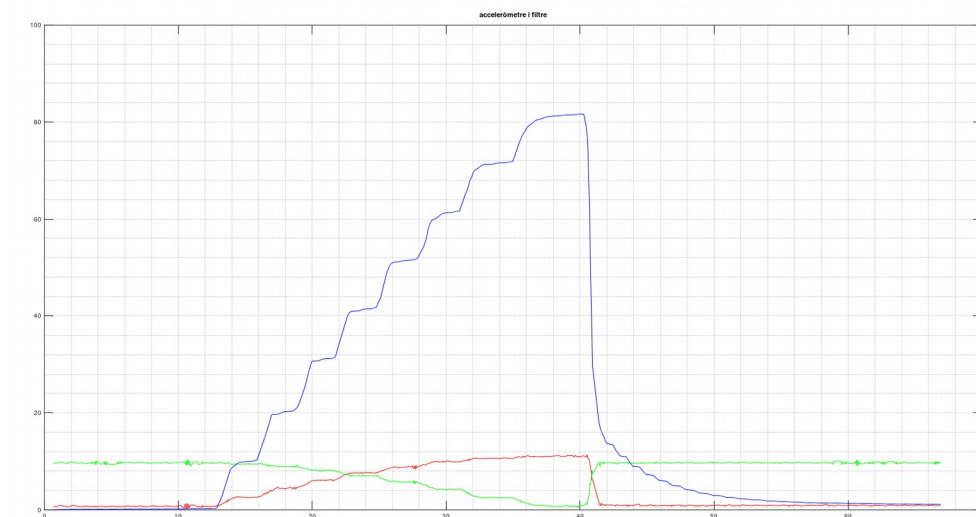


Figura 56: Gràfica 1 acceleròmetre i filtre +80° -> 0°

En la figura 55, podem veure la gràfica de les acceleracions amb el filtre complementari, en aquest cas si que es pot apreciar un cert retard en cada una de les mesures, aquest retard arriba a ser de 1 segon per l'increment de 10° en 10° i de entre 10 i 15 segons per la mesura de 90° a 0°. Com es pot apreciar en la gràfica aquest retard seria massa gran per poder detectar efectivament les inclinacions d'una motocicleta ja que aquesta té uns moviments més ràpids, tot i això podria ser eficaç per la mesura de inclinacions de llarga durada, ja que en aquest cas podem apreciar molt menys soroll en les mesures comparant-les amb les mesures efectuades amb l'algorisme de ponderació del 50%.

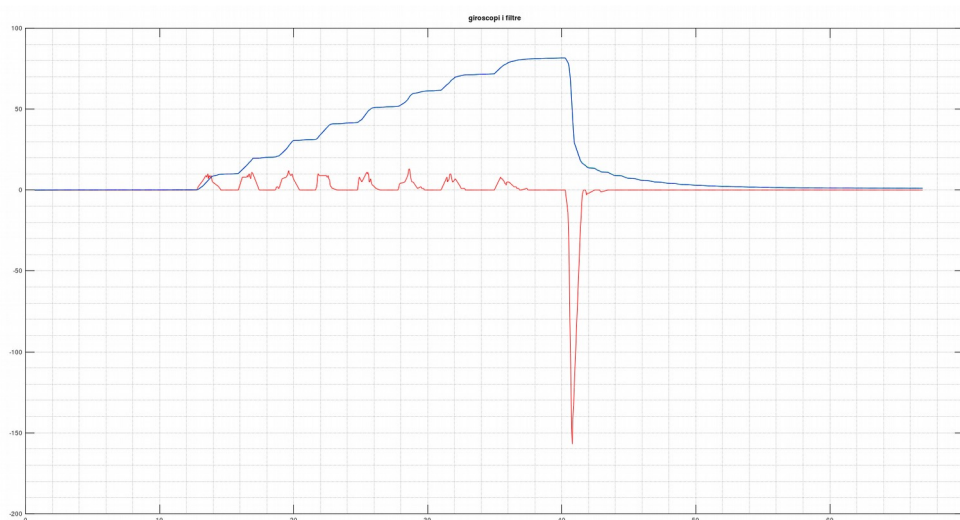


Figura 57: Gràfica 2 giroscopi i filtre +80° -> 0°

En la figura 56 podem veure que la velocitat de gir en els casos de l'increment dels 10° en 10° és molt lenta, contràriament el retorn a 0° l'efectuem ràpidament i es en aquest que podem apreciar més retard en la mesura. Per tant podem deduir que els girs ràpids afecten negativament al giroscopi i li incrementen el drift que aquest pateix, per tant aquest acaba amb un error superior al normal el qual haurà d'arreglar l'acceleròmetre, però aquest al tindre tant poca ponderació triga molt de temps a poder estabilitzar aquest error.

2.3.5 Prova 5 Determinació dels valors del filtre complementari a 85% giroscopi, 15% acceleròmetre.

En aquest apartat hem determinat que pel que hem pogut veure en els apartats anteriors hem de determinar un filtre amb un valor superior al 50% per el giroscopi per tal de mitigar el soroll generat per l'acceleròmetre, però que a la vegada sigui prou ràpid per tal de poder reaccionar al sistema en el qual el volem posicionar. Per tant determinem un percentatge del 85% pel giroscopi i un 15% per l'acceleròmetre en el filtre complementari.

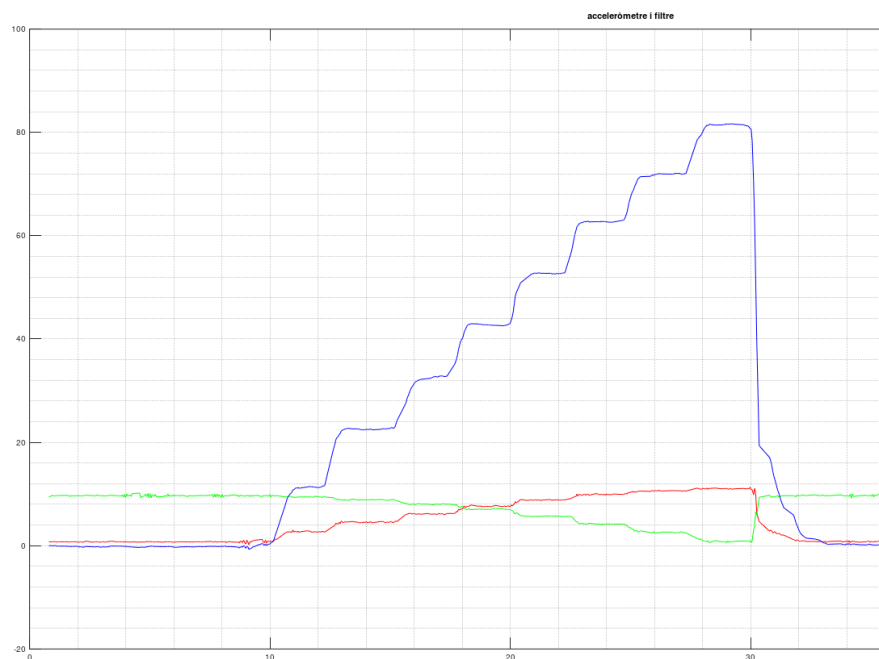


Figura 58: Gràfica 1 acceleròmetre i filtre +80° -> 0°

En la figura 57 podem veure les inclinacions determinades per el filtre amb un increment de 10° en cada un d'ells fins arribar a 80° i el retorn d'aquests 80° fins a 10° i posteriorment el retorn suau a 0° . Podem veure que en aquest cas l'algoritme és més ràpid, ja que podem determinar l'angle del sistema amb aproximadament 0.6 segons. Tot i això per depèn de quines aplicacions encara hauria de ser més ràpid però consideram que per detectar una

caiguda és suficientment ràpid amb 0.6 segons cada 10° . En la tornada de 80° a 10° aquest torna molt més ràpid ja que el retornem amb més rapidesa. Per tant la velocitat de càlcul també depèn molt de la rapidesa del canvi efectuat tal i com es pot apreciar amb les gràfiques de les acceleracions en vermell i verd.

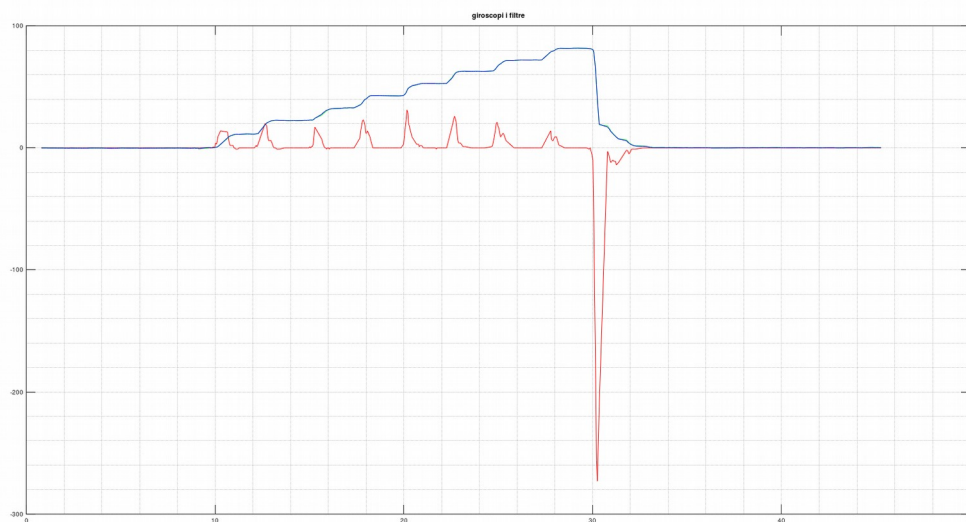


Figura 59: Gràfica 2 giroscopi i filtre +80° -> 0°

Tal i com podem veure en la figura 58 a partir del gràfic de color vermell, la velocitat en la rotació de cada 10° és molt petita per tant és normal que l'algoritme trigui més en establir aquella inclinació si hi està havent una progressió en aquesta, s'ha de tindre en compte que estem tenint una velocitat de moviment en aquesta prova de aproximadament $15^\circ/\text{s}$ en els increments de 10° en 10° . Contràriament la acceleració que li apliquem al sistema en el retorn és molt més alta aproximadament de $-270^\circ/\text{s}$ per tant la velocitat de retorn en aquest cas és molt major. A part d'això el soroll en les lectures és pràcticament menyspreable.

Per comprovar el soroll que tenim en les lectures extraïem una mostra de valors a 0° i calculem una mitjana i una variació estàndard de les mostres obtingudes. Obtenim una població de mostres durant 25 segons i obtenim:

| | |
|--------------------|-------------|
| Població | 385 mostres |
| Mitjana | -0,2° |
| Variació estàndard | 0,0674° |

2.3.6 Prova 6 Utilització del sistema en la motocicleta determinació dels angles a baixa velocitat en l'aparcament de casa.

Un cop preparat el prototip, és hora de provar-lo sobre la motocicleta. Fent alguna prova amb l'acceleròmetre a 4G veiem que en aquest apareixen moltes interferències degut a les vibracions de la pròpia maquina. Per evitar aquestes vibracions es posa l'acceleròmetre a 16G.

```
void GY_85::SetAccelerometer()
{
    //Put the ADXL345 into +/- 4G range by writing the value 0x01 to the DATA_FORMAT
    register 0x31.
    Wire.beginTransaction( ADXL345 );    // start transmission to device
    Wire.write( 0x31 );                  // send register address
    Wire.write( 0x03 );                  // send value to write 16G
    Wire.endTransmission();              // end transmission

    //Put the ADXL345 into Measurement Mode by writing 0x08 to the POWER_CTL register.
    Wire.beginTransaction( ADXL345 );    // start transmission to device
    Wire.write( 0x2D );                  // send register address //Power Control Register
    Wire.write( 0x08 );                  // send value to write
    Wire.endTransmission();              // end transmission
}
```


2.3.6.1 Prova de les vibracions en parat.

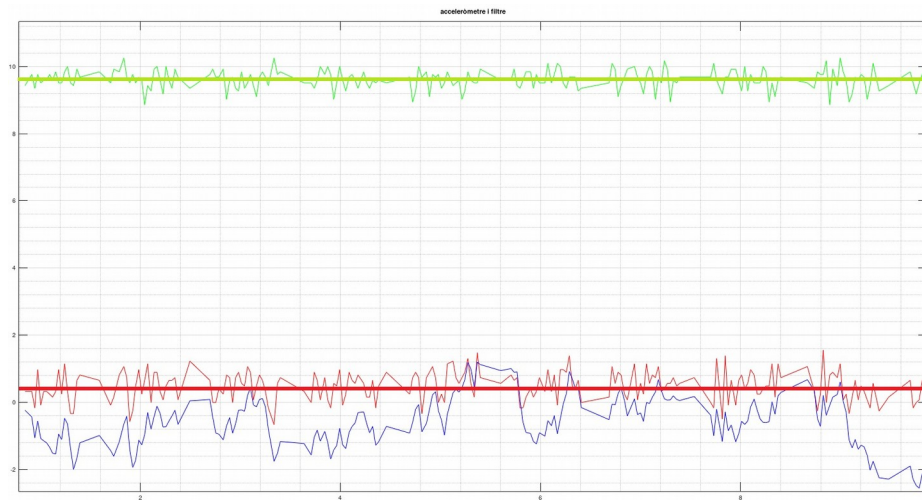


Figura 60: Soroll de l'acceleròmetre 4G amb l'aparell sobre la motocicleta sense moviment però amb el motor en marxa.

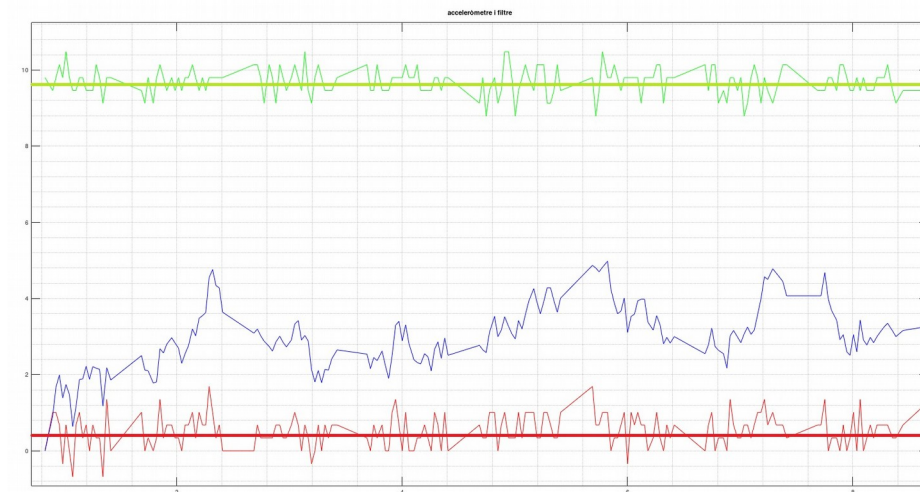


Figura 61: Soroll de l'acceleròmetre 16G amb l'aparell sobre la motocicleta sense moviment però amb el motor en marxa.

En la figura 59 podem veure una gràfica del soroll que genera la pròpia vibració de la motocicleta en l'acceleròmetre estant aquest calibrat a 4G. Per tal de minimitzar aquest efecte es calibra el sensor a 16G. Tot i això, com es pot apreciar en la gràfica 60, on podem veure el soroll que genera la vibració de la motocicleta en l'acceleròmetre, però aquesta vegada estant calibrat a 16G, no podem apreciar una gran millora en quan els valors. Sembla que aquests es moguin en els mateixos valors, fins hi tot podem arribar a concloure que en el de 4G s'observen els valors de les gràfiques més estables.

2.3.6.2 Prova en línia recta.

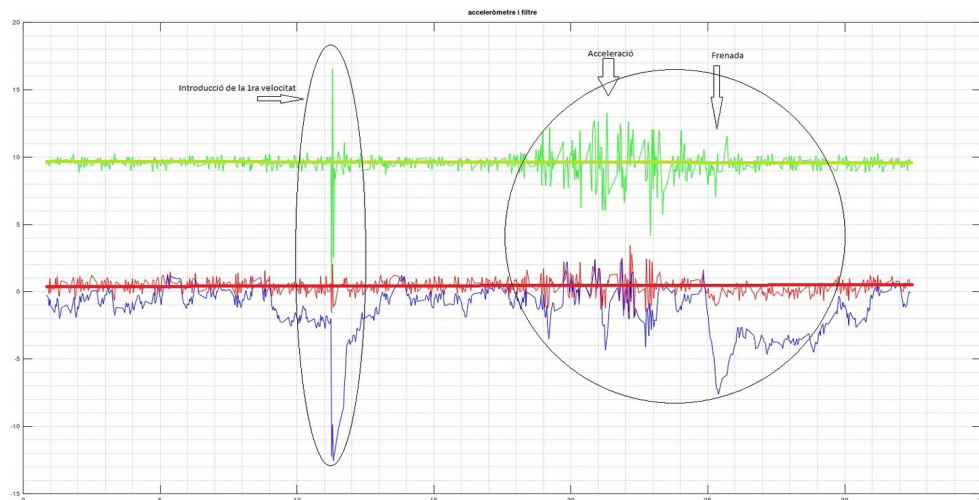


Figura 62: Acceleració i frenada amb la motocicleta. Acceleròmetre a 4G.

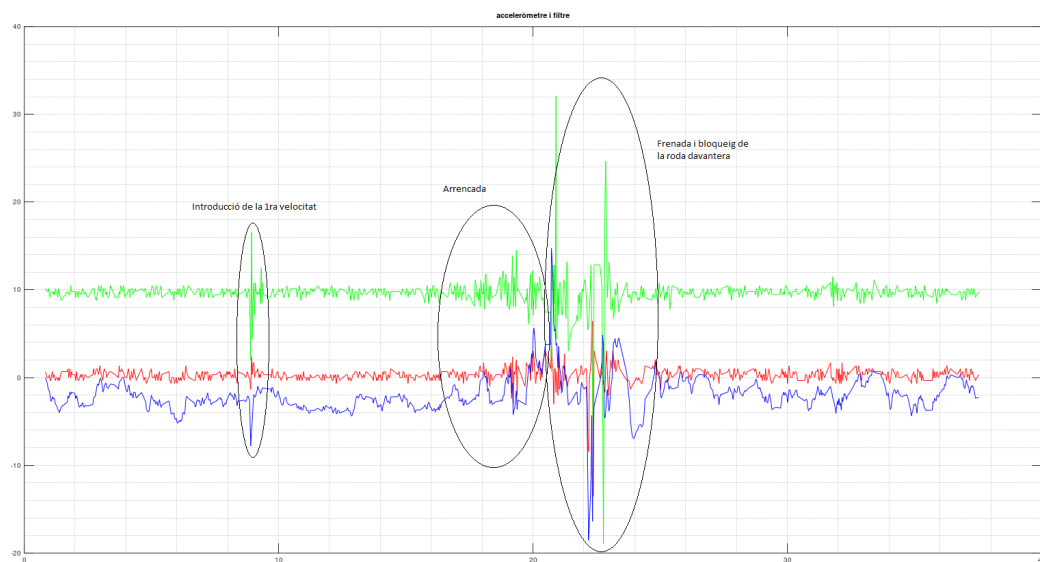


Figura 63: Acceleració i frenada amb la motocicleta. Acceleròmetre a 16G.

En la prova de la línia recte podem apreciar que quan introduïm la primera velocitat aquesta genera un soroll en l'eix Z prou gran com per que aquest modifiqui el valor de l'angle, en el cas del 4G fins a uns 12° i en el cas dels 16G fins a uns 8° . En quan a l'acceleració, en ambdós casos aquesta crea soroll que detecta l'acceleròmetre, en el cas del 4 G podem apreciar com aquest és més notori que en el cas dels 16G. A més a més en el cas dels 4G la prova s'ha portat a terme amb una acceleració suau i una frenada suau, i en el cas dels

16G l'acceleració ha sigut pràcticament igual, però en la frenada s'ha patit un error de pilotatge, s'ha bloquejat la roda davantera patint un tancament de la direcció havent de deixar anar els frens per tornar a frenar progressivament fins a aturar la motocicleta. Aquesta frenada ha estat captada per l'acceleròmetre i aquest ha donat uns valors molt grans afectant a la inclinació detectada.

2.3.6.3 Prova revol de dreta dins el garatge.

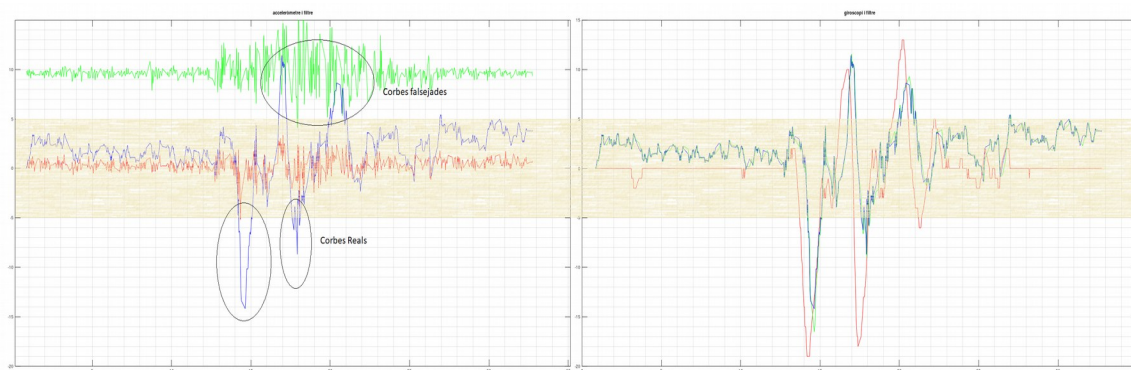


Figura 64: Doble corba a la dreta, acceleròmetre a 4G i giroscopi.

En la figura 63 podem veure la corba traçada dins el garatge, aquesta corba és una doble corba a la dreta en la qual s'ha de tornar a recuperar la inclinació entre un vèrtex i l'altre de la corba. En aquesta gràfica podem apreciar la inclinació cap a la dreta a partir dels nombres negatius, i la inclinació cap a l'esquerre a partir dels nombres positius. En la gràfica de l'esquerre es mostren les acceleracions, i en la gràfica de la dreta la velocitat de rotació a partir del giroscopi.

En el gràfic podem apreciar que hi ha aproximadament 4° de soroll, però podem considerar que aquests graus en una motocicleta és pràcticament a anar recte i això també queda camuflat per la baixa velocitat de la prova, ja que en aquestes baixes velocitats s'ha d'anar jugant amb la motocicleta per tal de poder mantenir l'estabilitat en aquest cas la corba s'ha traçat aproximadament entre 20 i 30 km/h. I segons la gràfica ens diu que hem inclinat 14° en la primera corba i 8° en la segona corba. Entre les dues corbes de dretes veiem que hi ha una corba falsejada d'esquerres, i que aquesta torna a reproduir-se després de la segona corba. S'entén que aquestes corbes són les recuperacions de les inclinacions de tombat a recte, però que per un error de la mesura dona una inclinació cap a l'altre banda, cal dir que aquesta és notòria ja que la gràfica marca 11° en la primera corba falsa, i 9° en la segona d'aquestes.

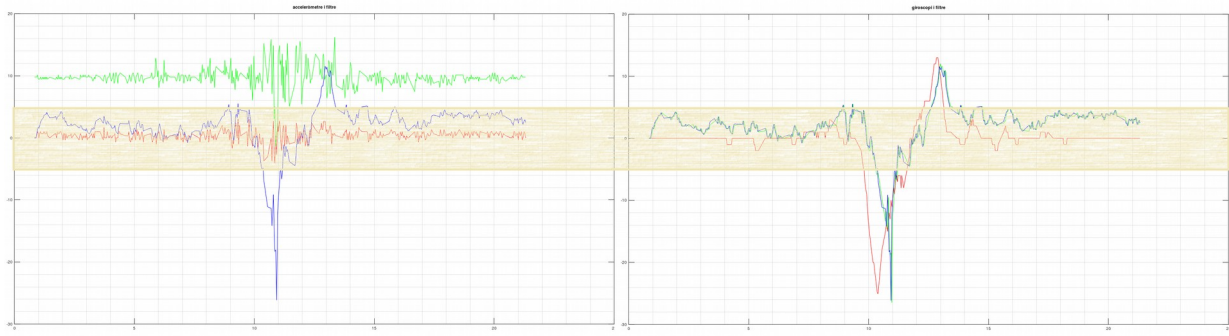


Figura 65: Corba a la dreta, acceleròmetre a 16G i giroscopi.

En la figura 64, podem veure la inclinació que s'ha mesurat amb l'aparell estant calibrat l'acceleròmetre a 16G. En la part esquerra podem veure la gràfica amb les acceleracions, i en la part dreta podem veure la gràfica amb la velocitat de rotació mesurada per el giroscopi.

En aquest cas la prova només l'hem fet amb una corba, la mateixa que s'ha fet per la gràfica anterior, però en aquest cas no s'ha fet el doble vèrtex, en aquest cas ens hem aturat després de la primera corba. Així doncs, en la gràfica podem veure la inclinació obtinguda durant la corba, aproximadament 25° . En aquest cas podem tornar a apreciar el falseig de la lectura del retorn en la inclinació de l'aparell, ja que en cap moment hem patit cap inclinació cap a l'esquerra. En aquest cas, també podem apreciar el soroll en la mesura dels 0° , tot i que en aquest cas també es queda dins la franja de 5° descrita anteriorment.

2.3.7 Prova 7 caiguda en parat.

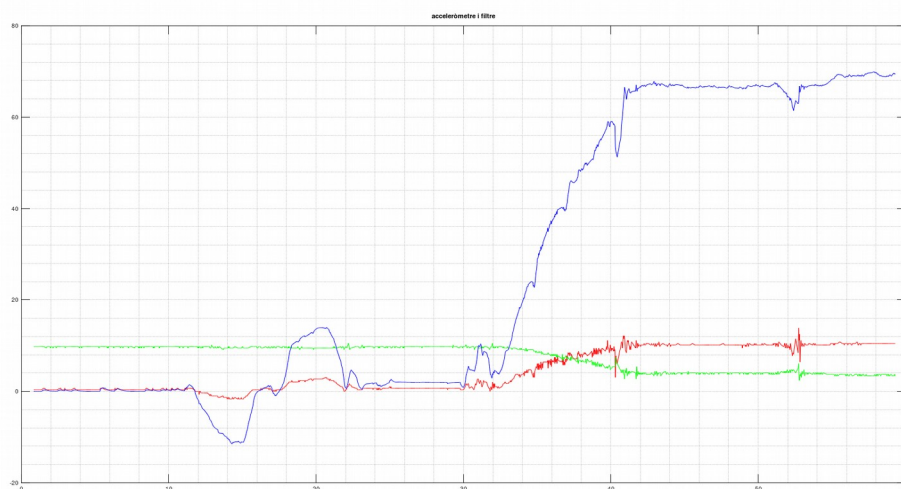


Figura 66: Valors de l'acceleròmetre en la caiguda, velocitat=0Km/h.

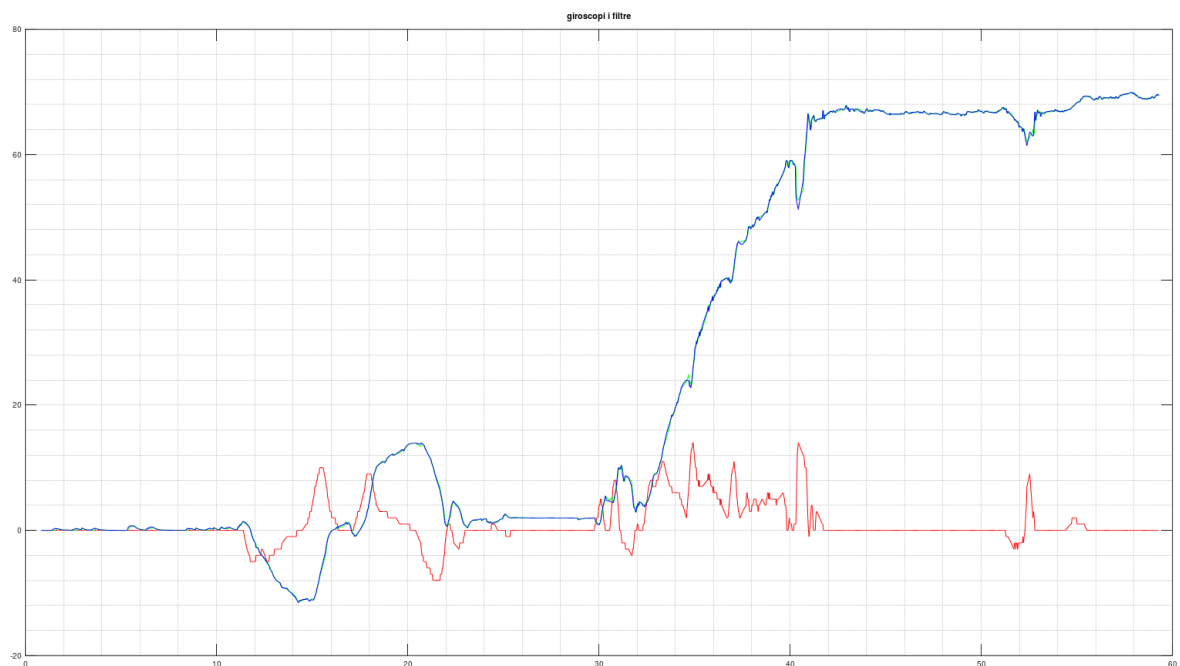


Figura 67: Valors del giroscopi en la caiguda, velocitat= 0Km/h.

En les figures 65 i 66, podem observar una caiguda en parat, aquesta s'ha efectuat suaument, tal i com mostren els valors del giroscopi i el temps de la caiguda en general. Per provar una mica l'aparell en parat ens movem una mica a dreta i esquerre, tal i com es pot apreciar entre els segons 18 i 22. En el segon 30, comencem a inclinar la moto cap a l'esquerre fins a deixar-la al terre. Un cop l'aparell està reposant tombada al terre aturem la mesura. Podem veure a partir de la gràfica que el valor màxim que pot arribar aquesta motocicleta són 68°. Per tant posant el limit de la caiguda als 65° en tenim suficient per aquest aparell.

2.3.8 Prova 8 dinàmica amb velocitat, traçat d'una rotonda, canvi de sentit.

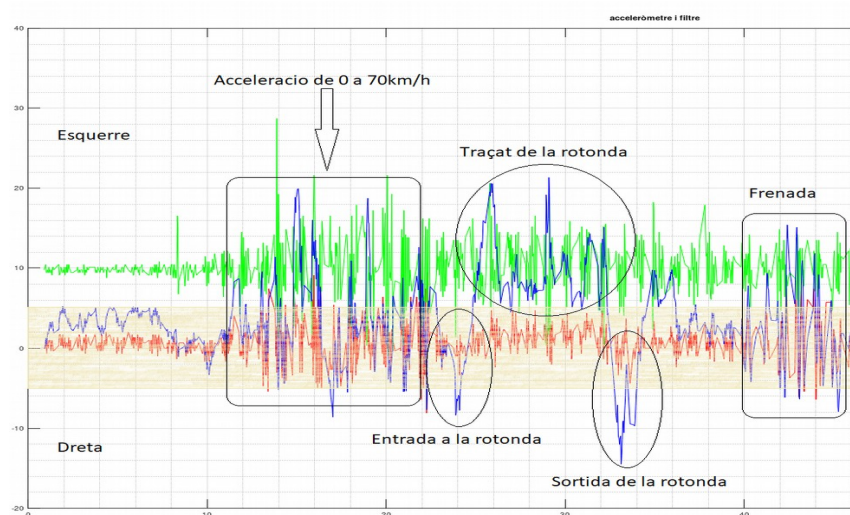


Figura 68: Gràfica d'acceleracions i inclinacions en el traçat de una

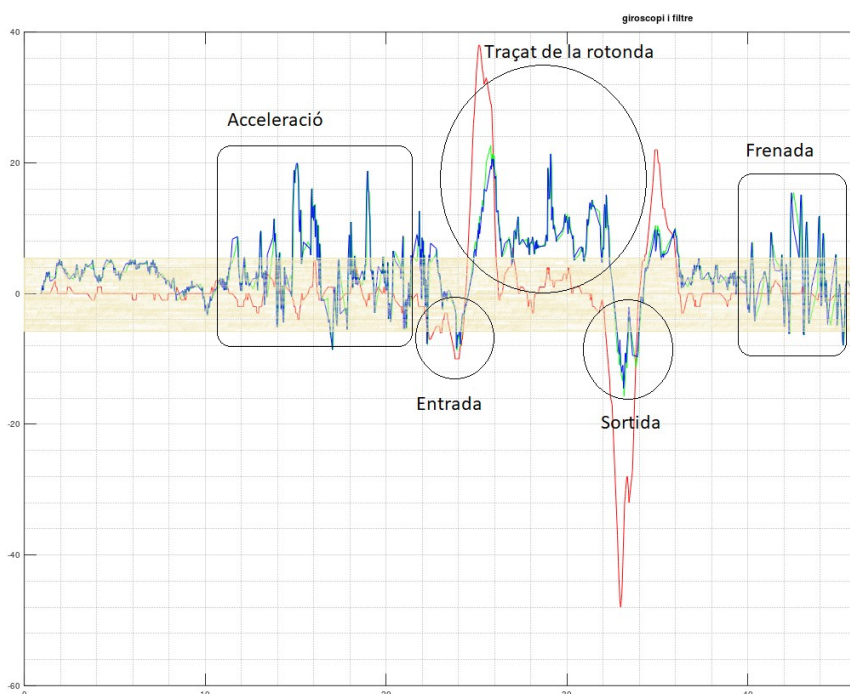


Figura 69: Gràfica de la velocitat de rotació i inclinacions en el traçat de una rotonda, canvi de sentit.

En les figures 67 i 68 podem veure els valors que dona l'aparell en fer el traçat d'una rotonda. En la prova, es comença de parat i s'accelera la motocicleta fins aproximadament 70km/h, un cop assolida la velocitat d'aproximació, es desaccelera i s'agafa la rotonda

aproximadament a 55 km/h i 3ra velocitat. En la rotonda s'efectua un canvi de sentit per tant 360° aproximadament sobre aquesta. Com que ens trobem a Espanya, les rotondes es tracen cap a l'esquerre, per tant en la gràfica podem veure l'entrada de la rotonda la qual és una petita inclinació cap a la dreta de uns 8° seguit de un gir llarg a l'esquerre, és en aquest punt on ens trobem traçant la rotonda. En el traçat d'aquesta anem variant la inclinació de la motocicleta per poder mantindre l'equilibri, i segons el gràfic no passem dels 20°. Seguidament sortim de la rotonda amb un gir a la dreta el qual podem apreciar a la gràfica 68 ja que en aquest gir de sortida es canvia la inclinació molt ràpidament i és en aquest punt on el giroscopi detecta una velocitat de rotació molt elevada. A partir d'aquí seguim recte fins que frenem i ens aturem.

Si observem la gràfica 67 detingudament, veurem que en la zona de l'acceleració i la frenada l'acceleròmetre és molt poc precís, ja que en aquests punts detecta acceleracions provinents del canvi de velocitat de la màquina i ens processen aquestes com a inclinacions en l'aparell quan realment si ens fixem en la figura 68 en aquell punt pràcticament no apreciem grans canvis en la rotació detectada per el giroscopi.

Per tant la conclusió a la que arribem és que en dinàmic l'acceleròmetre ens està perjudicant els càlculs de les inclinacions ja que aquest es veu influenciat per acceleracions no previstes per el sistema les quals afecten aquest negativament en forma de soroll i de falses lectures.

2.3.9 Prova 9 Localització de la caiguda.

Per tal de poder localitzar el lloc on s'ha efectuat la caiguda, el prototip incorpora un mòdul GPS. Per les proves que s'han efectuat amb aquest, cal dir que és un sistema lent en quan a la localització. Per tal de localitzar el lloc, en les nostres proves hi havia vegades que aquest era incapaç de trobar la nostre localització, i si la trobava era després de aproximadament 5 minuts. També cal dir les proves s'han efectuat en diversos dies de l'any i en localitzacions diferents, tant en ciutat, com a poble, dins de casa com a fora el carrer. Les vegades que el sistema ha tingut més problemes és dins de la ciutat, dins de un pis. En aquest cas, el sistema ha sigut incapaç de trobar la nostre localització. S'ha de dir que quan la prova s'ha fet al poble, o fora el carrer, aquest no ha tingut problemes per localitzar la nostre posició, tot i això si que ha necessitat una estona considerable a establir la posició en la qual ens trobem. Sobretot aquest temps d'espera en la localització dels satèl·lits és molt gran quan l'aparell porta una estona aturat. O quan aquest es reinicia. quan aquest ja ha estat consultat triga molt menys a localitzar la posició.

| COM11 | |
|-----------------|---|
| 14:33:48.812 -> | 0.03 |
| 14:33:48.879 -> | 0.02 |
| 14:33:48.913 -> | 0.02 |
| 14:33:49.013 -> | 0.02 |
| 14:33:49.485 -> | 0.31 |
| 14:33:49.586 -> | 0.26 |
| 14:33:49.620 -> | 0.22 |
| 14:33:49.654 -> | 0.19 |
| 14:33:49.722 -> | 0.16 |
| 14:33:49.756 -> | 0.14 |
| 14:33:49.789 -> | 0.12 |
| 14:33:49.823 -> | 0.10 |
| 14:33:49.857 -> | 0.08 |
| 14:33:49.890 -> | 0.07 |
| 14:33:49.958 -> | 0.06 |
| 14:33:50.395 -> | -68.47 |
| 14:33:50.565 -> | Alarma |
| 14:33:51.003 -> | Satellit: 5, Localitzacio: 41.5485954284 , 2.1024899482 |
| 14:33:51.104 -> | Satellit: 5, Localitzacio: 41.5485954284 , 2.1024899482 |
| 14:33:51.173 -> | Satellit: 5, Localitzacio: 41.5485954284 , 2.1024899482 |
| 14:33:51.340 -> | Satellit: 5, Localitzacio: 41.5485954284 , 2.1024899482 |
| 14:33:51.410 -> | Satellit: 5, Localitzacio: 41.5485954284 , 2.1024899482 |
| 14:33:51.545 -> | Satellit: 5, Localitzacio: 41.5485954284 , 2.1024899482 |
| 14:33:51.579 -> | Satellit: 5, Localitzacio: 41.5485954284 , 2.1024899482 |
| 14:33:52.018 -> | Satellit: 5, Localitzacio: 41.5485954284 , 2.1024916172 |
| 14:33:52.086 -> | Satellit: 5, Localitzacio: 41.5485954284 , 2.1024916172 |
| 14:33:52.186 -> | Satellit: 5, Localitzacio: 41.5485954284 , 2.1024916172 |

Figura 70: Localització GPS en ciutat.

Com es pot veure en la figura 69, en aquest cas no ha sigut necessari esperar-se gaire per tal d'obtenir 10 localitzacions, ja que el mòdul GPS havia estat utilitzat feia poca estona. Tot i això, com ja es diu en el primer paràgraf d'aquest punt, quan el mòdul GPS fa estona que no funciona, aquest necessita una bona estona per trobar els satèl·lits i proporcionar-nos una localització. En la primera inicialització del dia, en aquest cas ens hem agut d'esperar aproximadament 5 minuts.

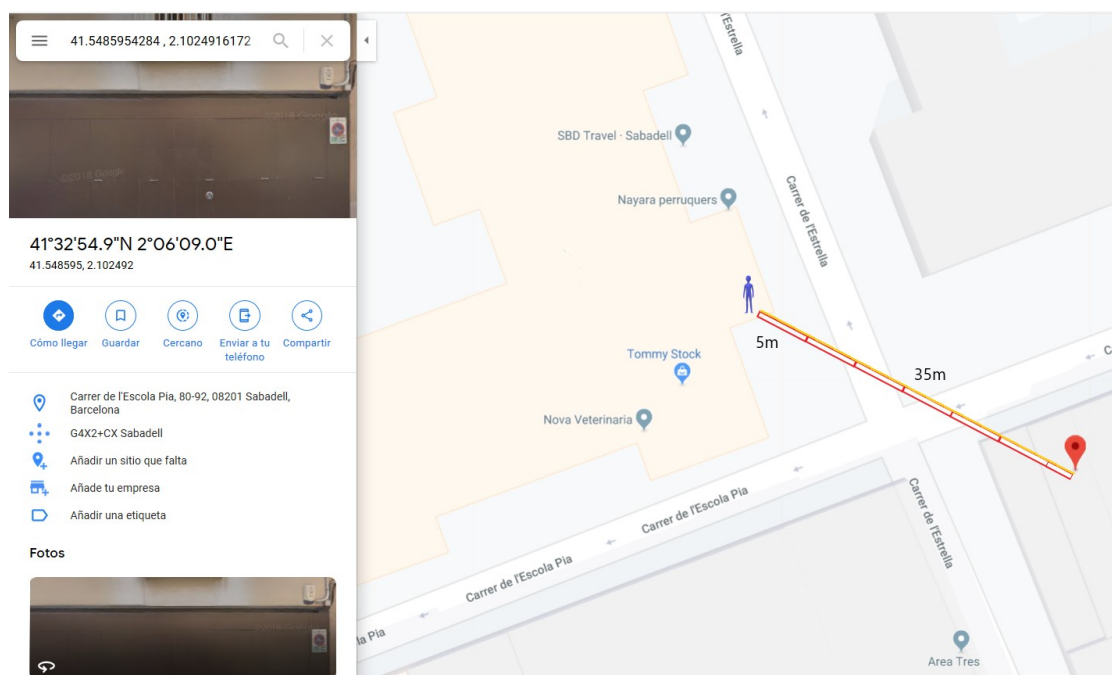


Figura 71: Localització GPS en el Google Maps

En la figura 70 podem veure la nostre localització des de on s'estan efectuant les proves amb la personeta de color blau. I amb el punter vermell podem veure la localització que ens està proporcionant el prototip fabricat. El Google Maps ens proporciona un petit regla en la part inferior que en aquest cas ha estat eliminat en la imatge, però hem aprofitat aquest per fer les mesures amb els rectangles vermells, d'aquesta forma tenim una referencia amb la qual podem mesurar. En aquest cas la referencia és de 5 metres tal i com es veu en la figura 70. Sumant totes referencies d'amidaments trobem l'error en distància que ens dona el GPS. En aquest cas aquest error és de 35 metres. En el programa de localització, en el GPS li estem demanant 10 decimals.

2.4. Fitxa tècnica de l'aparell.

| | |
|-----------------------------------|---|
| Alimentació | USB 5V |
| Consum | 280 mA |
| Mides | 97mm x 80mm x 61mm |
| Eixos de mesura | 1 eix |
| Connectivitat | Bluetooth LoRa |
| Localització | GPS |
| Error en la localització | +/- 35m |
| Temps en la localització | 1 ^r cop màxim 5 min Localitzacions posteriors menys de 30 s |
| Emmagatzematge de dades | Micro SD |
| Error màxim en la mesura estàtica | 3° |
| Error màxim en la mesura dinàmica | 10° |
| Graus de mesura | -65° a 65° |
| Activació de l'alarma | Valor > 65° Valor < -65° |

3. ESTUDI ECONÒMIC

Econòmicament el projecte seria viable a gran escala i portant a terme un prototip 2. Aquest prototip 1 no seria viable econòmicament ja que aquest és un conjunt de mòduls ja fabricats (plaques petites) que per elles mateixes ja fan una funció determinada. A partir d'aquest prototip es podria dissenyar un prototip 2 arreglant les mancances del prototip 1 i incorporant ja tots els components en la mateixa placa.

- Cost del prototip 1:

| Mòdul | Cost en € per unitat |
|--|----------------------|
| Microcontrolador Pro Micro, Atmega32U4, 5V, 16MHz. | 7,99 € |
| IMU Gy-85 9 DOF. | 7,45 € |
| LoRa AS32 TTL 100. | 14,80 € |
| Bluetooth HC-05. | 8,49 € |
| GPS GY-NEO6MV2. | 17,90 € |
| Modul lector targetes micro SD. | 3,99 € |
| Fabricació del circuit imprès. | 10 € |
| Total cost material per 1 prototip. | 70,62 € |

En aquest punt només es fa referència al cost de fabricar 1 prototip. Però realment el cost ha sigut molt major, ja que s'ha comprat més components per errors que haguin pogut tindre alguns d'aquests mòduls. Per exemple el mòdul que hem agut de substituir més és el GPS ja que aquest és sensible a l'electricitat estàtica. Del microcontrolador en tenim 2 unitats per tal de poder fer proves fora del prototip. De mòdul LoRa també s'ha adquirit 2 unitats per tal de poder construir un receptor.

A la vegada de que no s'han comptabilitzat els costos en personal que ha requerit el desenvolupament del projecte, ja que en aquest cas només s'ha desenvolupat per una sola persona, però això ha tingut un cost en temps molt elevat aproximadament de 250 hores de feina. Si contéssim a aquestes hores a 15 € la hora el qual és un preu raonable per la feina de

un enginyer obtindríem un cost en personal de 3750 €. Cal dir que aquest projecte es va començar a desenvolupar al Febrer del 2019 i s'està finalitzant amb aquest treball a l'Octubre del 2019.

4. CONCLUSIONS

4.1 Conclusions sobre l'aparell fabricat.

Les conclusions que podem extreure del treball fet fins ara són les següents:

- En aquest treball s'han assolit alguns dels objectius plantejats en la introducció, però alguns altres no s'han pogut assolir. Les proves efectuades demostren que l'aparell requereix d'alguns ajustos per poder realitzar la feina que se li encomana el 100% bé.
- L'aparell construït funciona correctament com a inclinòmetre remot a partir de les tecnologies sense fils incloses en aquest. Bluetooth i LoRa.
- Els resultats obtinguts de l'aparell en les proves estàtiques són correctes, aquest ens proporciona la inclinació entre -90° i 90° de forma efectiva i amb un error considerat com a petit.
- Els resultats obtinguts de l'aparell en les proves dinàmiques no són correctes del tot, ja que aquest té errors per els motius següents:
 - L'aparell té soroll a 0° degut a la vibració produïda pel motor de la motocicleta.
 - L'aparell és sensible a accions com l'acceleració de la màquina i la frenada d'aquesta, tot i que bona part d'aquestes forces no passen en els eixos de mesura de la inclinació.
 - L'aparell detecta el retorn de la motocicleta a posició 0° com si aquesta inclines cap al costat contrari al qual havia inclinat anteriorment.
- Tot i les complicacions descrites anteriorment, concloem que l'aparell si que pot ser utilitzat com a detector de caigudes en la motocicleta, ja que les inclinacions mesurades en la prova dinàmica de la rotonda en cap cas superen els 20° havent portat la motocicleta a una inclinació normal.
- L'aparell ens permet localitzar-lo a partir de coordenades GPS quan en aquest ha succeït alguna situació d'emergència.
- L'aparell té una bona connectivitat per incloure's en el món de l'IOT, ja que aquest ens permet connectar amb ell a partir de la tecnologia Bluetooth la qual ens obre la porta a dispositius com els Smartphones i a partir d'aquest connectar-nos a internet.

- L'aparell també serveix com a aparell de telemetria a partir de la targeta micro SD. Aquesta ens permet guardar les dades i analitzar-les a posteriori per tant d'extreure conclusions del que ha succeït en l'aparell.

4.2 Futures línies de treball.

Aquest treball no és una conclusió per aquest projecte. Aquest ha de continuar, ja que en aquest treball s'ha efectuat una primera part d'aquest. Algunes futures línies de treball per aquest projecte són les següents:

- Creació d'una aplicació per Smartphones que extregui les dades obtingudes per Bluetooth. Aquesta aplicació ha d'analitzar les dades enviades fins al dispositiu i en cas d'emergència informar per alguna aplicació de comunicació o missatgeria de la situació del sistema i la localització d'aquest.
- Creació d'un segon prototip corregint les mancances d'aquest primer prototip com poden ser la capacitat de càlcul o modificar el sensor IMU per un que ens permeti un millor tractament de les acceleracions en dinàmic.

5. BIBLIOGRAFIA

Jimblom, J. I. M. B. L. O. M. Pro Micro & Fio V3 Hookup Guide. [en línia] Consulta 21 febrer, 2019, de <https://learn.sparkfun.com/tutorials/pro-micro—fio-v3-hookup-guide/all>.

Llamas, L. L. Luis. Usar un acelerómetro ADXL345 con Arduino. [en línia] Consulta 23 febrer, 2019, de <https://www.luisllamas.es/arduino-acelerometro-adxl345-hookup-guide/all>

DEVICES, A. D. ANALOG. (2008). *ADXL345 datasheet*. [en línia] Consulta de <https://html.alldatasheet.com/html-pdf/254714/AD/ADXL345/24/1/ADXL345.html>

SENSE, I. S. INVEN. (2010). *ITG 3205 datasheet*. [en línia] Consulta de <https://html.alldatasheet.com/html-pdf/1170719/TDK/ITG-3205/63/1/ITG-3205.html>

Llamas, L. L. Luis. Leer y escribir en una tarjeta SD o micro SD con Arduino. [en línia] Consulta 15 MARç, 2019, de <https://www.luisllamas.es/tarjeta-micro-sd-arduino/>

Llamas, L. L. Luis. Conectar Arduino por Bluetooth con los módulos HC-05 ó HC-06 [en línia] Consulta 20 Març, 2019, de <https://www.luisllamas.es/conectar-arduino-por-bluetooth-con-los-modulos-hc-05-o-hc-06/>

Cristobal, A. R. (2019, 30 Març). TUTORIAL MODULO AS32-TTL-100. [en línia] Consulta 30 abril, 2019, de <https://www.crisalctime.es/?p=1513>

Hart, M. H. Mikal. (2013). TinyGPS++. [en línia] Consulta 4 juny, 2019, de <http://arduiniiana.org/libraries/tinygpsplus/>

6. ANNEXOS

6.1. Annex 1. Muntatge del prototip a la motocicleta.



